FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Reliable Mobility Support in Low-Power Wireless Networks

Mohammad Hossein Fotouhi Ghazvini

DISSERTATION

Doctoral Program in Electrical and Computer Engineering

Supervisor: Prof. Mário Jorge de Andrade Ferreira Alves

April 16, 2015

Reliable Mobility Support in Low-Power Wireless Networks

Mohammad Hossein Fotouhi Ghazvini

Doctoral Program in Electrical and Computer Engineering

Approved by:

President: Prof. José Alfredo Ribeiro da Silva Matos
External Referee: Prof. Jorge Miguel Sá Silva
External Referee: Prof. José Alberto Gouveia Fonseca
FEUP Referee: Prof. Manuel Alberto Pereira Ricardo
FEUP Referee: Prof. Luís Miguel Pinho de Almeida
FEUP Referee: Prof. Francisco Vasques de Carvalho
Supervisor: Prof. Mário Jorge de Andrade Ferreira Alves

April 16, 2015

List of author's publications

This is a list of papers and publications that reflects the results achieved during the development of the research work presented in this dissertation.

Books

- N. Baccour, A. Koubâa, C. Noda, **H. Fotouhi**, M. Alves, H. Youssef, M. Zuniga, C. Boano, K.Roemer, D. Puccinelli, T. Voigt, and L. Mottola. Radio link quality estimation in low-power wireless networks. Springer Lecture Notes in Electrical and Computer Engineering, 2013.

Journals

- **H. Fotouhi**, M. Alves, M. Zuniga, A. Koubâa, Reliable and fast hand-offs in low-power wireless networks, IEEE Transactions on Mobile Computing, 2014.
- **H. Fotouhi**, D. Moreira, M. Alves, mRPL: boosting mobility in the Internet of Things, Elsevier Journal on Ad-hoc Networks, 2014.

Conferences and workshops

- **H. Fotouhi**, M. Zuniga, M. Alves, A. Koubâa and P. Marron, "Smart-HOP: A Reliable Handoff Mechanism for Mobile Wireless Sensor Networks", at 9th European Conference on Wireless Sensor Networks (EWSN), 2012.
- **H. Fotouhi**, M. Alves, A. Koubâa and N. Baccour, On a Reliable Handoff Procedure for Supporting Mobility in Wireless Sensor Networks, at 8th International Workshop on Real-Time Networks (RTN) in conjunction with the 22nd Euromicro International Conference on Real-Time Systems (ECRTS), 2010.

Posters and demos

- N. Baccour, M. Jamâa, A. Koubâa, M. Alves, H. Youssef, **H. Fotouhi**, D. Rosário and L. Becker, RadiaLE: a framework for benchmarking link quality estimators, Demo and Poster at 7th European Conference on Wireless Sensor Networks (EWSN), 2010.
- **H. Fotouhi**, M. Alves, A. Koubâa and M. Zuniga, Smart-HOP: A Reliable Handoff Procedure for Supporting Mobility in Wireless Sensor Networks, Poster at 8th European Conference on Wireless Sensor Networks (EWSN), 2011.

- **H. Fotouhi**, M. Zuniga, M. Alves, and A. Koubâa, Efficient Handoff Mechanism for Mobile Wireless Sensor Networks, Poster at Students Conference in Electrical and Computer Engineering Conference (StudECE), 2012.

Technical reports

- **H. Fotouhi**, M. Alves, M. Zuniga, and A. Koubâa, "Reliable and fast hand-offs in low-power wireless networks (extended paper), Technical Report, 2014.
- **H. Fotouhi**, D. Moreira, and M. Alves, mRPL+: a mobility management framework in RPL, Technical Report, 2014.

Papers under submission

- **H. Fotouhi**, D. Moreira, M. Alves, P. Meumeu, Mobility management in RPL (under submission to an international journal).

Abstract

The need for reliable mobility support in low-power wireless networks (LPWNs) is ever increasing in application contexts such as health-care monitoring, industrial automation, smart cities and generally in the "Internet of Things". The wireless nature, the environmental context and scalability features of these applications usually impose the use of low-power and low-cost devices. Namely, these devices feature simple single radio transceivers, communicating at very low TX/RX power and using basic electronics and antennas. These characteristics lead to unstable, asymmetric and unreliable wireless links, which in turn affect the quality of service of the LPWN. This quality of service is even more difficult to achieve when the applications at stake support nodes mobility, as in heath-care monitoring and preventive/corrective maintenance in industrial environments. In this context, this Thesis aims at guaranteeing reliable and timely communication in LPWN under nodes mobility.

The main research objectives of this Thesis are: (i) to devise a reliable and fast hand-off mechanism for LPWNs, using state-of-the-art (SOTA) commercial-off-the-shelf (COTS) technologies, (ii) to integrate the proposed hand-off mechanism within a SOTA/COTS protocol stack and (iii) to support the mobility solution with analytical, simulation and experimental evaluations.

Within this research context, we designed a hard hand-off mechanism (dubbed smart-HOP) based on averaged single link quality parameters — received signal strength indicator (RSSI) or signal-to-noise ratio (SNR). We fine-tuned the most relevant hand-off parameters — lower threshold level (T_ℓ), hysteresis margin (HM), window size (ws), stability monitoring (m) — with preliminary experiments. We studied the impact of radio channel parameters — path loss exponent and shadowing standard deviation — on hand-off performance (packet delivery ratio and hand-off delay) based on a probabilistic model. A simulation model was designed to further study the impact of link monitoring frequency (window size) and the stability monitoring parameters on hand-off performance. An extended experiment with a more realistic environmental set-up was performed to better fine-tune the hand-off parameters.

We integrated smart-HOP within the commodity RPL/6LoWPAN routing protocol (dubbed mRPL) in a simple, effective and backward compatible manner. We also developed a soft hand-off mechanism to seamlessly change parent nodes while moving. In order to enable soft hand-off, we activated neighbor overhearing mechanism to eliminate the Discovery Phase in the hand-off process. We applied the soft hand-off model on top of mRPL (mRPL+) to further increase network performance.

Overall, results indicate that mRPL and mRPL+ are able to provide network connectivity and responsiveness by switching between parents in a short period of time while

successfully delivering most of data packets. mRPL and mRPL+ are able to provide good reliability ($\approx 100\%$ packet delivery ratio) compared with RPL (delivering $< 50\%$ of data packets). The average hand-off delay in RPL is ≈ 3 s, while it reduces to 85 ms in mRPL and 4 ms in mRPL+.

Resumo

Tem havido um interesse e utilização crescentes das redes de comunicação sem fio de baixo consumo - *low-power wireless networks (LPWN)*, para aplicações tão diversas como monitorização ambiental, automação industrial, agricultura de precisão e na Internet de próxima geração (denominada de *Internet of Things*).

Os requisitos da maior parte destas aplicações, quer em termos de durabilidade, escalabilidade, contexto ambiental e execuibilidade económica, impõem a utilização de dispositivos sem fio de baixo custo e de baixo consumo energético. Desta forma, os dispositivos (normalmente denominados "nós") sem fio dispõem normalmente de um único rádio, comunicando com potências de transmissão/recepção muito baixas e utilizando circuitos e componentes básicos, nomeadamente no que toca aos circuitos de modulação/desmodulação e antenas.

As características acima referidas resultam em ligações rádio instáveis, assimétricas e pouco fiáveis, que por sua vez afectam a qualidade de serviço da rede. Esta qualidade de serviço fica ainda mais comprometida quando as aplicações suportam mobilidade de nós, como na monitorização de doentes em ambientes hospitalares ou domésticos ou na manutenção preventiva e correctiva de processos industriais.

Neste contexto, esta Tese endereça a garantia de comunicações fiáveis em *LPWN* com nós móveis. Os objectivos principais deste trabalho são os seguintes: i) a concepção de um mecanismo de comutação entre pontos de acesso rádio (*hand-off*), recorrendo a tecnologias normalizadas e comerciais; ii) integração do mecanismo de *hand-off* proposto numa pilha protocolar normalizada; iii) sustentar esta solução integrada recorrendo a uma avaliação baseada em modelos analíticos, de simulação e experimentais.

Consequentemente, concebemos um mecanismo de *hand-off*, denominado de *smart-HOP*, baseado na estimação da qualidade do sinal rádio (força do sinal - *RSSI* - e relação sinal/ruído - *SNR*), tendo efectuado um ajuste inicial do valor dos parâmetros de *hand-off* mais importantes - nível de sinal mínimo para desencadear o *hand-off* (T_ℓ), margem de histerese (*HM*), duração da janela (*ws*) e monitorização de estabilidade (*m*) - através de um modelo experimental preliminar.

Posteriormente, analisámos o impacto de dois parâmetros do canal rádio - *path loss exponent* e *shadowing standard deviation* - na performance do mecanismo de *hand-off*, através de um modelo probabilístico. Foi também concebido um modelo de simulação que nos permitiu estudar a influência dos parâmetros *ws* e m no comportamento do mecanismo de *hand-off*. Finalmente, um cenário experimental mais realista permitiu-nos optimizar os parâmetros de *hand-off*.

Em linha com o objectivo ii, integrámos o mecanismo *smart-HOP* com o protocolo *RPL/6loWPAN*, denominando-o de *mRPL*, de uma forma simples, eficaz e compatível

com o standard. Concebemos também um mecanismo de *hand-off* melhorado e que permite aos nós móveis uma comutação transparente e mais rápida entre pontos de acesso rádio, tendo integrado ambos os mecanismos de *hand-off* numa única framework, denominada de *mRPL+*.

Globalmente, verificámos que a framework proposta nesta Tese suporta a mobilidade de nós de forma muito mais eficaz que o protocolo *RPL standard*, garantindo tempos de desconexão negligenciáveis (na ordem dos milisegundos) e uma fiabilidade muito elevada (cerca de 100% dos pacotes são recebidos correctamente).

Acknowledgements

Foremost, I would like to express my sincere gratitude to my supervisor, Mário Alves, for his continuous support during my PhD study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this Thesis. I would like to thank him for giving me the opportunity of working in different projects, initiating interactions with external people and offering me the possibility of co-supervising his Master students and co-reviewing papers.

Besides my supervisor, I would like to thank Marco Zuniga (Delft University of Technology) and Anis Koubâa (Prince Sultan University) for their helps in building the framework of this work. The one month internship for doing research and experiments under Marco's supervision was very fruitful. I learned how to define problems, come up with solutions and more importantly how to present the work. I should also thank Nouha Baccour (University of Sfax) for providing nice ideas in leveraging link quality estimators for hand-off mechanism. I had also very fruitful interactions with Carlo Fischione (KTH Royal Institute of Technology) and José Ramiro (University of Sevilla) during my PhD research.

I would like to acknowledge the hosting under CISTER/INESC-TEC Research Center and the Polytechnic Institute of Porto (ISEP/IPP). I also thank my lab mates for the stimulating discussions and all the fun we had during these years. Stefano Tennina for his help in TinyOS implementation and Patrick Meumeu for useful comments in revising the analytical part of my work.

Finally, and most importantly, I would like to thank my dear wife, Maryam Vahabi, for her support, encouragement, patience and unwavering love. She made these years the best years of my life. I thank my beloved parents, Reza Fotouhi and Mina Mahmoudi, who raised me with love and supported me in all my pursuits.

Hossein Fotouhi

This work was supported by FCT (Portuguese Foundation for Science and Technology) and by ESF (European Social Fund) through POPH (Portuguese Human Potential Operational Program), under PhD grant SFRH / BD / 71823 / 2010.

Contents

List of Figures

List of Tables

List of Algorithms

Abbreviations and symbols

6LoWPAN	IPv6 over Low-power Wireless Personal Area Networks
AP	Access Point
ARSSI	Average Received Signal Strength Indicator
ASNR	Average Signal-to-Noise Ratio
CDMA	Code Division Multiple Access
COTS	Commercial-Off-The-Shelf
CTI	Cross Technology Interference
CTP	Collection Tree Protocol
DAG	Directed Acyclic Graph
DAO	Destination Advertisement
DAO-ACK	Destination Advertisement Object Acknowledgment
DD	Directed Diffusion
DIO	DODAG Information Object
DIS	DODAG Information Solicitation
DODAG	Destination Oriented DAG
FHSS	Frequency Hopping Spread Spectrum
F-LQE	Fuzzy Link Quality Estimator
GPS	Global Positioning System
GSM	Global System for Mobile Communications
HM	Hysteresis Margin
ICMPv6	Internet Control Message Protocol version 6
ISM	Industrial, Scientific and Medical
LPWN	Low-Power Wireless Network
LQE	Link Quality Estimator
LTE	Long Term Evolution
MAC	Medium Access Control
MASP	Maximum Amount Shortest Path
MIPv6	Mobile IPv6
MN	Mobile Node
OLSR	Optimized Link State Protocol
OS	Operating Systems
PCB	Printed Circuit Board
PDR	Packet Delivery Ratio
PRR	Packet Reception Rate
QoS	Quality-of-Service

RAM	Random-Access Memory
RF	Radio Frequency
RFID	Radio Frequency IDentification
ROM	Read-Only Memory
RPL	Routing Protocol for Low-Power and Lossy Networks
RSSI	Received Signal Strength Indicator
SCD	Stop to Collect Data
SNR	Signal-to-Noise Ratio
SOTA	State-Of-The-Art
TDMA	Time Division Multiple Access
TTDD	Two Tier Data Dissemination
UMTS	Universal Mobile Telecommunication System
WLAN	Wireless Local Area Networks
WS	Window Size
WSN	Wireless Sensor Network

Chapter 1

Overview

This introductory chapter describes the research topic, identifies the gap and problems to be addressed and states our hypothesis. Then, it outlines our approach and specific research objectives. We conclude the chapter with a summary of the main scientific contributions of the Thesis and with a brief description of how it is organized.

1.1 Research context

This Thesis addresses the support of mobility in low-power wireless networks. Within this context, this section outlines (i) the importance of low-power wireless networks, some commodity communication technologies and their main features and differences, (ii) the need for supporting mobility in many applications, the challenges and the existing protocols and platforms; and finally (iii) the existing techniques to handle mobility in such networks.

1.1.1 Low-power wireless networks

Low-power wireless networks (LPWNs) are becoming increasingly important for many applications, ranging from precision agriculture to industrial automation, smart cities and health-care. By 2020, there could be over 50 billion devices that communicate wirelessly [(IB14]. According to the Global System for Mobile Communications (GSM) Alliance, just a quarter of these devices will be mobile handsets and personal computers,

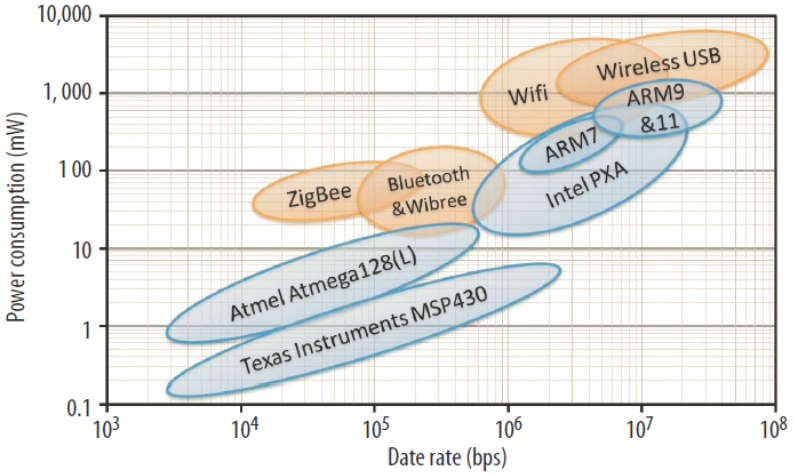


Figure 1.1: Average power consumption of wireless transceivers and microcontrollers in various wireless networks with respect to their data rates [YY06].

as most autonomous connected devices will operate and intercommunicate without user interaction mainly via LPWNs. These technologies include radio frequency identification (RFID) [Rob06], Bluetooth [Haa00], ZigBee [All06], IPv6 over Low-power Wireless Personal Area Networks (6LoWPAN) [SB11] and solutions based on sub-GHz technologies. Many LPWN standards such as IEEE 802.15.4 [ZL04], WirelessHART [SHM+08], ISA100.11 [ISA09], IETF 6LoWPAN [SB11], IEEE 802.15.3 [Fis07], Wibree [Hun06], and IEEE 802.15.1 [GT00] have been devised. A common feature of all these technologies and standards is low-energy consumption.

Low-power wireless networks operate at a very low data rate and transmission power aiming at a prolonged lifetime. Figure 1.1 shows a comparison of power consumption of wireless transceivers and microprocessors in different wireless systems. The maximum transmission power of a regular LPWN device (e.g. TelosB [CT14]) is 1 mW, while in WiFi access points (APs[1]) is in the range of 30 mW to 800 mW, in WiFi mobile nodes (laptops) is 32 mW, in cellular APs is $\approx 10^5$ mW and in cellular phones varies from 500 mW to 2 W[2].

A low-power radio link operates on a radio frequency (RF) power budget that rapidly drops by enlarging the distance between sender and receiver. The radio signal propaga-

[1]In wireless communications, a wireless AP is a device that allows wireless devices to communicate with the rest of the network, either through a wireless or wired infrastructure.

[2]The values related to transmit power in WiFi/cellular MNs and APs are not depicted in Figure 1.1.

tion is also affected by other factors that contribute to link quality degradation, such as: (i) multi-path propagation effect, (ii) the interference from the concurrent transmissions within or between wireless networks — cross technology interference (CTI) — and (iii) hardware transceivers that may distort transmissions.

Early LPWN hardware platforms such as ChipCon CC1000 and RFM TR1000 operate in sub-GHz frequencies. The low data rate of these devices may prevent their use in many data collection applications. The devices working in the 2.4 GHz ISM band, such as the ChipCon CC2420 and CC2500 families, enable higher data rates.

Complementing the radio hardware is the use of special antennas for the transmission over the air. LPWN devices are often equipped with cheap low-gain antennas integrated in the board. For instance, the antenna in a TelosB device is integrated in a printed circuit board (PCB) and is omni-directional, resulting in an irregular pattern due to the presence of the node circuitry and battery package close to the antenna. These aspects complicate the operation of Medium Access Control (MAC) and routing protocols, which are traditionally based on the assumption of uniform communication ranges (devices with same maximum transmission range) and symmetric links (bidirectional links with less than 10% difference in Packet Delivery Ratio (PDR) [SAZ10]).

1.1.2 Mobility support in LPWNs

Mobile applications. Low-power wireless technologies are important building blocks for an increasing range of applications requiring mobility support, such as industrial automation, health-monitoring and robotics [par14]. These application domains push low-power networks to dramatically improve Quality-of-Service (QoS) properties in terms of throughput, timeliness, reliability, security, privacy, usability, and efficiency [par14, Com14a, MMK12]. These QoS requirements must be guaranteed between mobile nodes and also between the mobile nodes and fixed network infrastructures.

A recent NSF report [par14] presents an exemplary scenario capturing this situation. In the future, in-body sensors collecting "*aggregated data from the entire population can predict outbreaks of epidemics even before they occur*" and it further states that "*such applications require in-body sensors that not only have robust wireless connectivity, but*

also are highly energy-efficient". We can easily foresee a hospital covered by a LPWN infrastructure (used for one or more purposes). Patients use body sensors to monitor relevant/vital signs — they are monitored and tracked either when standing still or moving (walking, wheel chair or in bed). Doctors and medical staff also use body sensors, which can be used to measure their stress/anxiety levels and also to track and warn them about emergency situations. Only if these wearable body sensor nodes are able to communicate reliably and in a timely manner, this will effectively transform medical services in the future.

In industrial environments such as factory automation and process control, it is essential to monitor the actual state of components and machines in a continuous manner. The Factories of the Future 2020 roadmap [Com14a] forecasts "*the need for advanced machine interaction with humans through ubiquity of mobile devices to receive relevant production information*". Such type of systems is also expected to detect potentially dangerous conditions in real-time and launch necessary countermeasures to prevent their impact on workers' health and safety.

The Cooperating Objects roadmap [MMK12] envisions several application domains requiring the cooperation between mobile robots instrumented with sensing/actuation capabilities with fixed low-power wireless nodes, such as for search & rescue, environment exploration and surveillance applications. A large number of small (and inexpensive) robots can cooperate to tackle a large problem. These swarms of robots pose important challenges to robot designers as their cooperative behavior is not as simple to program as a single robot: many algorithms are distributed and rely heavily on communication between the participating members of the swarm and also with a fixed wireless infrastructure. Typically, this communication is time-critical, meaning that it has to be completed within a deadline to be effective.

Many recent research projects (e.g. [GIN14, pm14, Com14b, MdDJGSBO14]) and research works (e.g. [SSB12, CLBR10a, LMFJ$^+$04a, LFNS12, PM11] have considered network architectures that require timely (or at least continuous) data collection from mobile nodes through low-power wireless interfaces to fixed network infrastructures. In oil refineries, workers are exposed to hazardous environments in highly critical ar-

eas, so collecting the workers' vital signs during their daily activity enables to quickly detect abnormal situations [SSB12]. In clinical monitoring, patients have embedded sensing devices that report real-time streams of information through a fixed infrastructure [LMFJ+04a, CLBR10a]. Mobile robots are also used to assist fixed sensor network deployments in wildlife monitoring to detect and extinguish fire [LFNS12, PM11].

In this Thesis, we consider mobility support within LPWNs, where mobile and fixed nodes coexist.

Interference. Low-power network protocols such as IEEE 802.15.4 operate in the 2.4 GHz Industrial, Scientific and Medical (ISM) band, which is shared by IEEE 802.15.1 (Bluetooth) and the far more powerful IEEE 802.11b/WiFi. To make things worse, most commodity wireless devices (e.g. baby monitors, walkie-talkies, and microwave ovens) use the same 2.4 GHz ISM band. Since all these systems coexist in the ISM wireless spectrum, tackling the external interference generated by other wireless systems is paramount.

IEEE 802.15.4 and 802.11b operate in the same frequency spectrum, but they occupy different frequency channels. In our experiments, we tune the LPWN IEEE 802.15.4 nodes to communicate on Channel 26 to reduce the probability of interference from WiFi nodes, as this channel is largely immune to 802.11b interference [SDTL10].

Bluetooth is based on frequency hopping spread spectrum (FHSS) technology. It uses 79 different 1 MHz-wide channels. Experiments in [SDTL10] show that the interference from Bluetooth can be as high as 25 dBm, i.e. on the same order as for 802.11b networks. Due to the frequency hopping (every 625 µs), the resulting packet loss may be lower than the one deriving from 802.11b interference.

1.1.3 Major concepts and definitions

A common solution to keep network connectivity in mobile wireless networks is for mobile nodes to perform *hand-offs*. **Hand-off (or hand-over) is the process where mobile nodes select the best access point (AP) available to communicate.** The major concepts related to hand-off in LPWNs are presented next.

Soft or hard hand-off mechanism? The type of hand-off mechanism is dictated by the capabilities of the radio, communication protocols and applications. Hand-offs are classified into two main categories: *hard hand-offs* and *soft hand-offs*.

In a *soft hand-off*, the new link is established before disconnecting the current link, so the mobile node must be equipped with multiple radios. This characteristic enables a mobile node to communicate with several APs and assess their links quality while keeping normal communications with the serving AP. Nevertheless, cost, power and size restrictions usually preclude the use of multiple radios in LPWNs. It is possible to perform soft hand-off by utilizing a network-based mobility management, as the one supported by mobile IPv6. However, the use of IPv6 imposes extra overhead and drastically increases the energy consumption of the network, which is not practical in most LPWN applications [CPR+08, JSS+10].

In a *hard hand-off*, a mobile node drops the current link before searching for a new link; the radio can use only one channel at a given time, and hence it needs to stop the data transmission before the hand-off process starts. Consequently, in hard hand-offs, it is central to minimize the time spent looking for the best available AP. LPWN nodes typically rely on low-power radio transceivers that can operate on a single channel at a time, such as the widely used CC2420. However, it is feasible to devise an effective/efficient hard hand-off process for LPWNs, as we propose in this Thesis.

The hard/soft hand-off design can follow a standard model or a heuristic model. The standard models are usually employed in cellular, wireless mesh, wireless local area networks (WLANs) and 6LoWPAN networks. These standard models are based on the mobile IPv6 (MIPv6) mobility management mechanism, which is initiated by predicting node mobility according to the link and network information. In heuristic models, several input parameters influence the hand-off decision. Dynamic programming is an adaptive strategy that suites for modeling hand-off in mobile wireless networks that have a time varying context [FAS14]. Genetic algorithms are used for finding the exact or approximate optimization solution [CKMT02], while artificial intelligence techniques mimic the properties of biological systems to model the hand-off process [CB04]. Fuzzy logic is a multi-valued logic that can be used to combine several metrics to optimize the hand-off

decisions [FAKB10, ZVC14]. However, the aforementioned heuristic models introduce time and processing complexity that make them extremely challenging for LPWNs, as we figured out in the initial phase of our work [FAKB10].

Broadcast or hand-off? A naive solution to support mobile nodes in LPWN would be for mobile nodes to broadcast information to all APs within their wireless range. The broadcast approach, while simple, has a major limitation: it leads to redundant information at neighboring APs (since several APs receive the same packets). This implies that the fixed infrastructure has to either waste resources in forwarding the same information to the end point, or it imposes the use of a complex scheme, such as data fusion, to eliminate duplicated packets locally.

A more efficient solution is for mobile nodes to use a single AP to transmit data at any given time. This alternative requires nodes to perform hand-offs between neighboring APs. Hand-offs have been extensively studied in other wireless systems [CJC09, WL97, MHT00, SF06, MDT08, RS05, MSA03, SMA04], in particular cellular and WLANs. However, these techniques are not suitable for LPWNs. Contrarily to more powerful systems, such as WiFi and cellular networks, which have advanced spread spectrum radios and almost unlimited energy resources, LPWNs typically have severely constrained radio, processing memory and communication resources.

Transitional region in LPWNs. In LPWNs, links are usually classified according to their packet reception ratio. A transmitter/receiver node pair, where no packet is correctly received, is referred to as to have a *dead* link. A transmitter/receiver node pair, where $\leq 10\%$ of the packets are received is a *poor* link. A transmitter/receiver node pair, where $10-90\%$ of the packets are correctly received is an *intermediate* link, defining what is known as *transitional region*. Finally, A transmitter/receiver node pair where $> 90\%$ of the packets are received is a *good* link [SDTL10] — see Figure 1.2.

In order to study low-power links, we gathered received signal strength indicator (RSSI) and signal-to-noise ratio (SNR) values at different parts of a building [FZA+12]. Figure 1.3 depicts the three aforementioned regions, which match previous studies (e.g. [SKAL08]). The SNR is calculated by measuring the noise floor immediately after receiving the packet, and then subtracting it from the RSSI value (in dBm). The RSSI regions

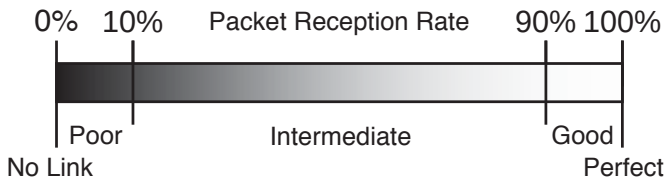


Figure 1.2: Link quality regions in LPWNs. A link is *dead* if it has a packet reception ratio (PRR) of 0%. A link is *poor* if the PRR is less than 10%, is *intermediate* if the PRR is between 10% and 90%, is *good* if the PRR is between 90% and 100% [SDTL10].

can be mapped directly to the SNR ones by subtracting the average noise floor.

Ping-pong effect. Low-power links have two characteristics that are more affected upon mobility: short coverage and high variability [ZIH+11]. Short coverage derives from a low density of access points. In high-power networks (e.g. cellular networks), a mobile node is likely to be within the range of many APs; this permits the mobile node to be conservative with link quality thresholds during a hand-off and to select very reliable links. On the other hand, in LPWNs, APs may not be deployed in such high densities, and hence the link quality thresholds should be more relaxed. In practice, this implies that the thresholds should be more carefully calibrated (within the unreliable transitional region, as we are going to see in this Thesis).

The high variability of links has an impact on stability. When not designed properly, hand-off mechanisms may degrade the network performance due to the *ping-pong* effect, which consists in mobile nodes having consecutive and redundant hand-offs between two APs due to sudden fluctuation of their links quality. This usually happens when a mobile node moves in the frontiers of two APs. Hence, to be stable, a hand-off mechanism should calibrate the appropriate thresholds according to the particular variance of its wireless links.

The transitional region for the IEEE 802.15.4 CC2420 radio transceiver encompasses the approximate range [-92 dBm, -80 dBm] (shown in Figure 1.3). Intuition may dictate that the closer the hand-off is performed to the connected region, the better (because links are more reliable). In practice, a hand-off starts when the link with the current (serving) AP drops below a given value (T_ℓ: lower threshold level) and stops when it finds a new AP with the required link quality (above T_h: higher threshold level). Figure 1.4(a) depicts this

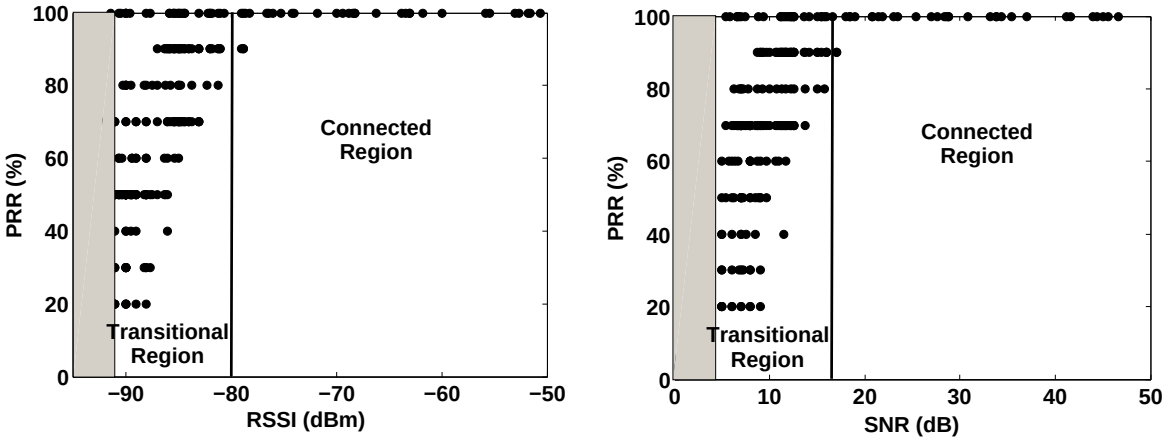
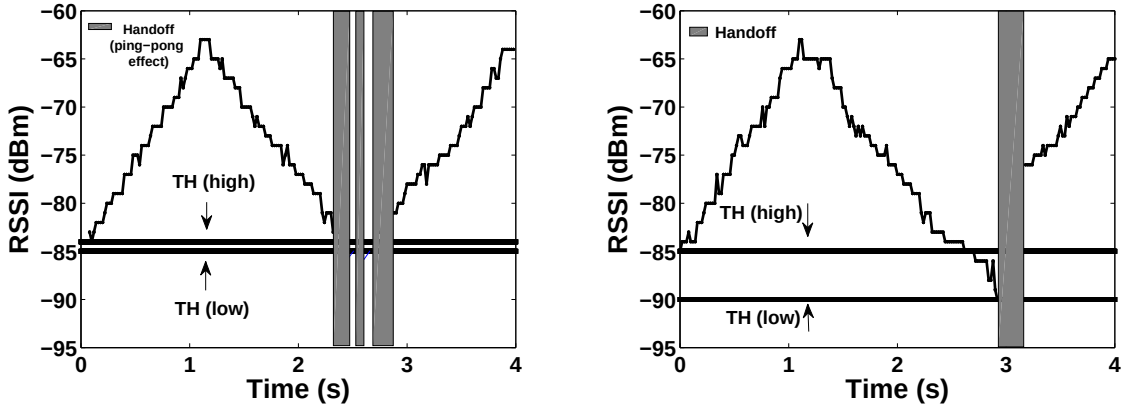


Figure 1.3: Low-power link model (a) RSSI vs. PRR. For RSSI greater than -80 dBm, the PRR is greater than 90%, and for RSSI less than -92 dBm, the PRR is less than 10%. In between, a small variation in the RSSI can cause a big difference in the PRR, which is identified as transitional region. (b) SNR vs. PRR. The borders for SNR are 4 dB and 16 dB, which are obtained by subtracting the noise floor from the RSSI readings [FZA+12].

conservative approach [FZA+12]. It considers -85 dBm as the lower threshold, and the upper threshold is 1 dBm higher. These parameters lead to a negative effect: a long hand-off delay (≈ 0.7 s) deriving from three consecutive hand-offs between the two contiguous APs (ping-pong effect). Figure 1.4(b) shows that by considering a wider margin, deeper into the transitional region, the ping-pong effect disappears and the hand-off delay reduces to approximately 0.2 s.

According to the low-power link features, we conclude that the hand-off process should be triggered when the link between the MN and AP is in the transitional region.

Link quality estimation. Radio link quality estimation is one of the key issues in a hand-off mechanism, as dynamic changes of low-power links quality require an accurate and fast estimation of the link quality. Selecting a proper link quality estimator requires studying the characteristics of dynamic, unreliable and variable wireless links upon mobility; in mobile LPWN applications, a good link quality metric is essential to a reliable, timely and energy efficient system operation. However, harsh environments with rapid variations of wireless channel quality turn it difficult to estimate the instantaneous value of the wireless link quality.



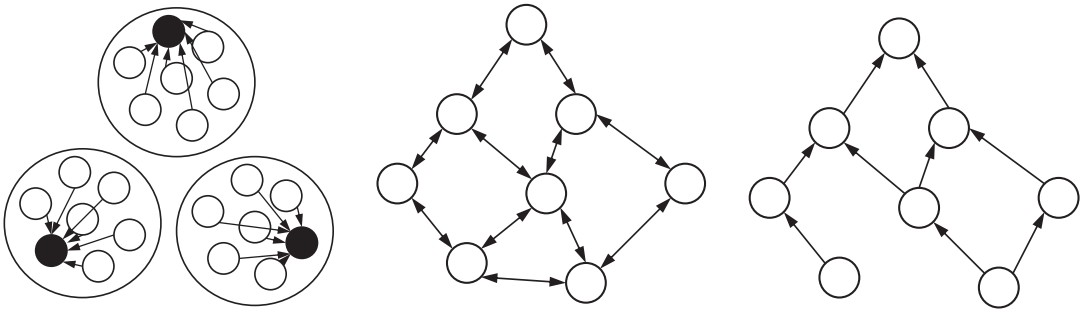
(a) (b)

Figure 1.4: (a) an example of an inefficient hand-off (ping-pong effect) with narrow hysteresis margin (1 dBm), $T_\ell = -86$ dBm and $T_h = -85$ dBm. (b) an example of an efficient hand-off with wide hysteresis margin (5 dBm), $T_\ell = -90$ dBm and $T_h = -85$ dBm [FZA+12].

Thus, it is generally defended that the RSSI or SNR indicators alone are not suitable for determining the quality of wireless links [BKY+10]. State-of-the-art link quality estimators (LQEs) combine a number of parameters to estimate the status of the link. However, we argue that complex LQEs are unpractical for LPWNs with mobile nodes. This statement was also confirmed in some other works [SL06, FGJL07, BVD+09].

We actually devised a fuzzy-based hand-off model at an early stage of the PhD, combining F-LQE [BKY+10] (a fuzzy-based LQE) with additional network parameters such as energy level, traffic load, and depth level [FAKB10]. However, after initial experimental tests using TelosB platforms, we found that such a complex heuristic-based hand-off model is not suitable for low data rate and low processing devices. Nevertheless, we believe that accurate heuristic-based hand-off models may turn to be effective in more powerful devices, such as Shimmer [BGM+10], Egs [KWS+10], Opal [JKK+11] and Iris [Inc14].

Bottomline, in applications with more stringent timing requirements, which run over current low-power and low-cost wireless hardware platforms, multi-criteria hand-off decisions lack responsiveness and accuracy. Thus, link quality estimators based on a single metric (such as RSSI and SNR) are preferred.



(a) (b) (c)

Figure 1.5: Three main routing paradigms in LPWNs: (a) cluster-based, (b) flat-based, and (c) tree-based.

In this Thesis, we assume a single link quality metric (RSSI/SNR) for estimating the radio link quality.

Mobility support in a routing paradigm. Routing mechanisms are devised according to the data gathering strategy, which we assume to be data collection from mobile nodes. We categorize the routing protocols into three main groups: (i) *cluster-based*, (ii) *flat-based*, and (iii) *tree-based*, which are briefly described as follows.

In a *cluster-based* routing protocols, the network is divided into clusters, as depicted in Figure 1.5(a). Within each cluster, a special node is responsible for collecting, aggregating and forwarding the information toward the data sink. Some examples of cluster-based routing protocols are LEACH [HCB02], PEGASIS [LR02], and HEED [YF04]. A mobile node within a cluster interacts with fewer nodes. However, adapting time-slots in common TDMA-based clustering approaches during mobility is very costly. Also, mobility between clusters adds extra overhead on cluster-head communications. Therefore, for a data collection application with mobile nodes, cluster-based routing is inefficient.

In *flat-based* routing, all nodes are considered equal (shown in Figure 1.5(b)). This architecture is useful when devices have the same resource capabilities. Some examples of flat-based routings are SPIN [HKB99], MCFA [YCLZ01], and AODV [PR99, PBRD03]. In flat-based routing, the route is created on demand by broadcasting control messages before data transmission. Relying on message broadcasts in a mobile network with resource limitations increases network overhead and energy consumption. Moreover, a flat-based network is not scalable, which makes it unsuitable for some applications.

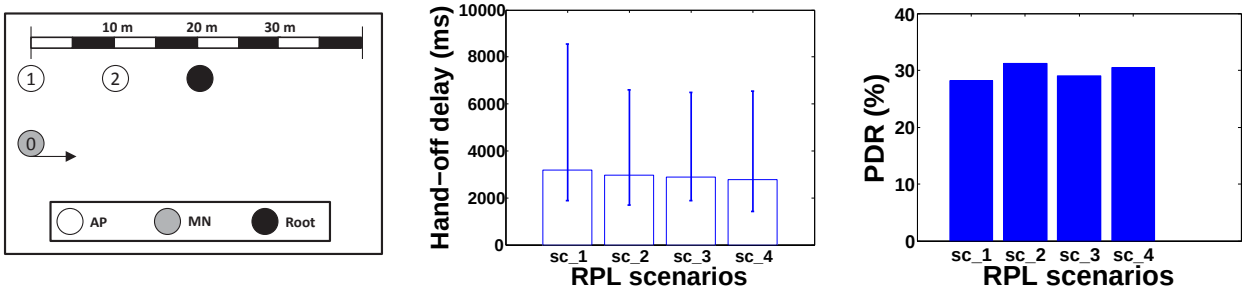
Figure 1.6: A network architecture of a mobile application, consisting a fixed set of nodes with mobile nodes.

In *tree-based* routing, a tree-like structure is created towards a sink node that collects data — see Figure 1.5(c). Collection Tree Protocol (CTP) [GFJ+09, FGJ+06] and Routing Protocol for Low-Power and Lossy Networks (RPL) [Win12, CHP11] are commodity tree-based routing protocols. Each node in the network has one parent and possibly one or more child nodes. The nodes in the network communicate with each other by exchanging an objective function, which is used to select and optimize routes (to the sink), based on the transmission cost. This feature makes tree-routing more suitable for LPWNs, but conventional tree-based routing protocols have been designed for static networks. However, we believe that tree-based routing can easily be adapted for supporting mobility, as we show in this Thesis.

In this Thesis, we will integrate a hand-off mechanism within a tree-based routing protocol (RPL).

1.2 Problem statement and hypothesis

Most LPWN protocols provide mechanisms to overcome topology changes due to joining and leaving of nodes, but the process of detecting node changes and updating routing tables is very slow. Figure 1.6 exemplifies the network model we use in this Thesis. It depicts hospital room, equipped with fixed sensor nodes, while patients, nurses and doctors (with sensors attached on their body) move in this environment. Mobile nodes



(a) (b) (c)

Figure 1.7: A snapshot of the impact of mobility in RPL routing; (a) network architecture with two static nodes (parents), a mobile node and a root node, (b) hand-off delay with different RPL settings, and (c) packet delivery ratio with different RPL settings.

generate traffic, which are then forwarded to the destination via a fixed node infrastructure (APs).

In order to show the problem in supporting mobility in state-of-the-art (SOTA) commercial-off-the-shelf (COTS) protocols, we performed tests with RPL, a de-facto standard routing protocol for IP-based networks (6LoWPAN). We used two static nodes and one mobile node, where the latter moved between the two static nodes — depicted in Figure 1.7(a). The nodes' transmission power was set to -25 dBm, while the mobile node generated packets at a 30 pkt/s rate. We determined the hand-off delay based on the time that the mobile node was disconnected from either parents. We considered four different scenarios, according to different RPL parameters setting. The results in Figure 1.7(b) show that the average hand-off delay varies from 2.8 to 3.2 s, which reveals the low responsiveness of RPL upon mobility. Long disconnections reduce data delivery during node mobility, so the packet delivery ratio is around 30% — see Figure 1.7(c). Hence, the native topology management mechanism in RPL is not adequate for supporting reliable and timely data communications upon nodes mobility.

In this Thesis, we address the following problems:

1. Reliable and fast data delivery during a hand-off by tackling the ping-pong effect (as illustrated in Figure 1.4).

2. Reliable and fast data delivery in a multi-hop and COTS routing protocol (RPL) between moving nodes and a fixed wireless infrastructure (as depicted in Figure 1.7).

Thesis hypothesis. Based on the aforementioned reasoning and problem statement, we formulate our hypothesis: *we believe that it is possible to improve the reliability and timeliness of standard/COTS-based LPWNs upon nodes mobility without extra hardware and with a minimal overhead by designing appropriate hand-off and re-routing mechanisms.*

1.3 Research objectives and approach

The primary objective of this Thesis was **to achieve reliable and timely mobility support in low-power wireless networks.** To reach this primary objective, a set of scientific and technical objectives have been identified.

- Devise a reliable and fast hand-off mechanism for LPWNs, using SOTA/COTS technologies.
- Integrate the hand-off mechanism within a SOTA/COTS routing protocol.
- Support our findings with analytical, simulation and experimental models.

In order to reach the aforementioned objectives, our approach was:

- Investigating the impact of mobility in candidate SOTA/COTS LPWNs technologies.
- Analyzing best LQE under mobility.
- Identifying a system/network model, considering a certain application pattern.
- Devising a hand-off mechanism without extra hardware/overhead and that performs reliably and fast.
- Integrating this hand-off mechanism in a SOTA routing protocol, trying to keep backward compatibility and with no or negligible changes to the standard protocol.
- Supporting and validating our proposals with analytical, simulation and experimental models.

1.4 Research contributions

We outline the scientific and technological contributions of the Thesis as follows:

1. We designed, implemented and validated a hard hand-off process, dubbed as **smart-HOP**:
   - at the initial phase of the PhD, we analyzed the feasibility of a more sophisticated link quality estimator for supporting hand-off decisions — the fuzzy link quality estimator [BKY+10];
     - published as a demo at EWSN 2010 [BJR+10] and a paper at RTN 2010 [FAKB10].
   - we devised smart-HOP, a hard hand-off process for LPWNs, based on IEEE 802.15.4 radios;
     - published a poster at EWSN 2011 [FAKZ11] and a paper at EWSN 2012 [FZA+12].
   - we implemented smart-HOP in TinyOS over COTS platforms, i.e. TelosB and AdvanticSYS xm1000 (preliminary experiments and extended experiments).
     - code available online [Pro14c].
   - we developed probabilistic and simulation models for further analysis and fine-tuning of the smart-HOP algorithm;
     - published in the IEEE Transactions on Mobile Computing [FAZK14].
2. We designed, implemented and validated a mobility management mechanism for RPL/6LoWPAN:
   - we have integrated smart-HOP within RPL — mRPL;
     - published in the Elsevier Ad-hoc Networks Journal [FMA14a].
   - we implemented mRPL in the Contiki operating system (over TelosB motes amd Cooja).
     - code available online [Pro14a].

  - we designed, implemented and tested a combination of soft and hard hand-off mechanisms that improves the mRPL — mRPL+; we devised multi-hop add-ons such as traffic awareness, best route selection, route establishment and loop avoidance; we also devised hand-off features such as backward compatibility, collision avoidance, freshness and timers to increase hand-off efficiency.
    - this work has been submitted to an international journal.
  - we implemented mRPL+ in Contiki OS (over ContikiRPL).
    - code available online [Pro14b].

Besides the scientific papers already mentioned, we shared our knowledge and expertise with the community by co-authoring a book at the Springer Lecture Notes in Electrical and Computer Engineering series [BKN+13].

1.5 Structure of the Thesis

The remainder of this Thesis is organized in three parts;

- in *Part I (Chapters 2, 3 and 4)*, we discuss the design and validation of a reliable and fast hand-off for LPWNs. Chapter 2 defines major concepts on mobility and hand-off mechanism in LPWNs while referencing the related works. Chapter 3 describes the design and preliminary evaluation of the proposed hard hand-off mechanism (smart-HOP). Chapter 4 presents the analytical, simulation and evaluation, as well as a consolidated experimental evaluation in a more realistic environment (extended experiments).

- in *Part II (Chapters 5, 6 and 7)*, we provide an integrated mobility management solution for the RPL/6LoWPAN protocol stack. Chapter 5 gives some examples of related work on mobility support in IP-based LPWNs. Chapter 6 presents the hard hand-off integration within RPL (mRPL), together with simulation and experimental evaluations. The additional mechanisms such as timers, loop avoidance

and RPL settings are also described. Chapter 7 presents a soft hand-off mechanism (mRPL+) that improves mRPL. Importantly, we conceive a mobility management framework, where a multi-hop routing protocol is able to support both soft and hard hand-off models. Then we describe the multi-hop add-ons. An analytical model to compare the soft and hard hand-off-based solutions is provided, which is followed by simulation-based evaluations.

– and in *Part III (Chapter 8)* we conclude the Thesis by stating he major lessons learned and some open issues that may be addressed in the future.

Chapter 2

Fundamental concepts and literature review

This chapter defines major concepts on mobility, while providing insights into the network architecture that has been used to ground this PhD work. We also provide some motivation examples of LPWN applications with mobility requirements. Along the chapter, we identify the most relevant related works and the particularities of our model and approach.

2.1 Characterizing mobility

2.1.1 Mobility types

LPWNs are one of the most appealing technologies for the future. LPWN nodes are easily installed in places that may be inaccessible or too expensive for wired systems. LPWNs may experience changes in network topology due to several reasons, namely due to mobility. In general, mobility can be classified into (i) physical mobility and (ii) logical mobility, as elaborated next.

Physical mobility. Physical mobility, which is also known as *strong mobility*, stands for the actual movement of a node from one place to another [ASU05]. It can be considered as a deliberate movement of objects, vehicles or people with LPWN nodes attached. Similarly, it can occur when nodes are dragged by external forces such as wind,

water, or air. In some applications, physical mobility plays a key role. In clinical monitoring applications, nodes are attached to the body of patients [DNW07] and medical staff [CKPK+12] to monitor their condition and activities. Workers in disaster recovery scenes [LMFJ+04b], oil extraction and refinery areas [CDL08] may be granted augmented capabilities if equipped with mobile network connectivity. Physical mobility causes frequent topology changes that results in deterioration of link quality, route changes, transmission delay and packet error/loss.

Logical mobility. Mobility is not restricted to a change in the position of a node from one physical location to another. Topology changes may happen due to the adding/removing of nodes, hardware failure, radio link degradation (e.g. interference, environmental changes, obstacles) and exhausting batteries. In these cases, some nodes may become inaccessible to their neighbors. These topology changes are referred to as "logical mobility" or *weak mobility* [ASU05]. Most LPWN protocols are able to deal with slow changes in the network.

In this Thesis, we consider physical mobility.

2.1.2 Mobility patterns

The mobility pattern is the characterization of the movement of real-life objects, such as vehicles and people, in terms of dimension, speed, direction, group behavior and predictability. The following aspects are paramount in a mobility pattern:

- *Group mobility.* When a set of nodes are standing and moving together for a considerably long time, this is identified as group mobility. Exploiting group behavior improves the performance of radio communication by employing adequate strategies (e.g. clustering).
- *Dynamics.* Speed, acceleration and trajectory.
- *Dimension.* Linear, planar or three-dimensional movement.

Understanding the mobility pattern helps in improving the network performance through an appropriate network architecture and protocol design. In this subsection, we identify higher level mobility patterns. There are three main mobility patterns in LPWNs [HFB09].

– *Pedestrian mobility pattern.* It describes the motion characteristics of people, e.g. when sensor nodes are attached to the human body. This pattern is characterized by speed limits and obstacles. The movement in this pattern is considered to be two-dimensional, and may or may not show a group behavior [Sch06].

– *Vehicular mobility pattern.* It describes the movement of vehicles equipped with sensor nodes. Vehicles can communicate with each other conveniently by capturing traffic conditions and other information. Vehicle movement is usually one-dimensional and characterizes a group behavior at a high speed in an industrial environment [Sch06].

– *Dynamic medium mobility pattern.* It describes moving through a medium, such as wind, water or other fluids. This mobility can be one-, two- or three-dimensional, depending on the medium type.

We consider pedestrian mobility pattern, where the speed of mobile nodes is limited.

2.1.3 Mobility models

A mobility model is a formal mathematical description generalizing the characteristics of mobility patterns, which may be an essential building block in analytical and simulation-based studies. Mobility models are needed in the design of location management strategies and radio resource management in wireless networks. The choice of the mobility model has a significant effect on the obtained results. If the model is unrealistic, invalid conclusions may be drawn. Mobility models are generally categorized into (i) trace-based models, and (ii) synthetic-based models [RS09].

In a trace-based model, real-life mobility patterns are collected from a large number of participants for a long observation period. However, the real movement trajectory of mobile nodes is difficult to capture even when sufficient historical data are obtained and recurrent mobility patterns occur. Synthetic models, on the other hand, attempt to represent the behaviors of real-world mobile objects. However, they cannot produce a

precise description of mobile patterns. There are several synthetic mobility models, from which we highlight the following ones [DD13].

– *Entity mobility models.* This class of mobility models stands for random models that are independent of network topology and performance parameters. A typical example is the *Random Walk Mobility Model* [CBD02]. It expresses the mobility of a node as it travels from its current location to a new location within a predefined time period or distance, by randomly choosing a direction and speed. A node changes its direction and speed once the time expires or the maximum permitted distance is reached. It is proven that a random walk on a one- or two-dimensional surface returns to the origin with complete certainty [Wei10]. A similar model is the *Random Waypoint Mobility Model*, which includes pause time between changes in the direction and/or speed. Mobility is greatly affected by the pause time and speed of nodes; for example, a fast movement of nodes and a long pause time results in a more stable network than a slow movement of nodes and a short pause time [CBD02].

– *Group mobility models.* In these models, the movement of nodes with respect to other nodes is of primary interest [AKBD06]. For example, the *Exponential Correlated Random Mobility Model* creates a motion function to predict the new location of mobile nodes in the next time-slot, to mimic an erratic movement [CBD02]. In the *Nomadic Community Mobility Model*, a group of mobile nodes collectively roams from one place to another according to the location of a reference node [CBD02]. Individuals move randomly within their own spaces following a random entity mobility model. Unlike the Nomadic Community Mobility Model in which the mobile nodes share a common reference point, the *Column Mobility Model* requires a one to one mapping of the anchor and the mobile subjects [CBD02].

In our work, we consider the Random Walk mobility model in simulations and a real pedestrian mobility model by attaching mobile sensors to a human body in real experiments.

2.1.4 Mobile nodes functionality

The role of mobile nodes may impact the design of the network architecture and communication protocols. The main components of a LPWN are:

- *Source nodes.* These nodes are the source of information. Such nodes perform sensing as their main task. They may also route packets depending on the communication protocol architecture.
- *Sink node(s).* These nodes are destinations of the sensing information. The data is collected directly from a source node or indirectly by an intermediate node.
- *Special nodes.* These nodes act as intermediate nodes or gateways and may (or not) be employed to improve network performance.

The mobility feature might be involved in any of the above components. For instance, source nodes might be mobile and the sink nodes might be static, or vice versa. In a mobile network, at least one of these components is assumed to be mobile. The mobile elements are categorized as follows [DFDA11].

Relocatable nodes. These mobile nodes change their location to better characterize the sensing area or for improving network connectivity or energy-efficiency. The relocatable nodes are not supposed to collect data during their movement. These nodes change their location and consequently change the topology of the network. A network architecture containing relocatable nodes is depicted in Figure 2.1(a). There exist many examples of employing relocatable nodes for improving communication reliability and energy-efficiency [BGLA03]. In some other research works, these nodes are used for improving the sensing coverage [WY05]. In this case, the primary concern is avoiding coverage holes. In some other approaches, the relocatable nodes are ordinary nodes that are randomly changing their location and may generate and forward traffic [HE04].

Mobile data collectors. These nodes collect the data that are generated by source nodes. The moving feature is either on purpose for improving the network reliability or it is unpredictable. Mobile sinks have been extensively addressed in the literature [RWB08] — see Figure 2.1(b). In most of these works, source nodes are static and one or multiple mobile sinks collect the information from all nodes.

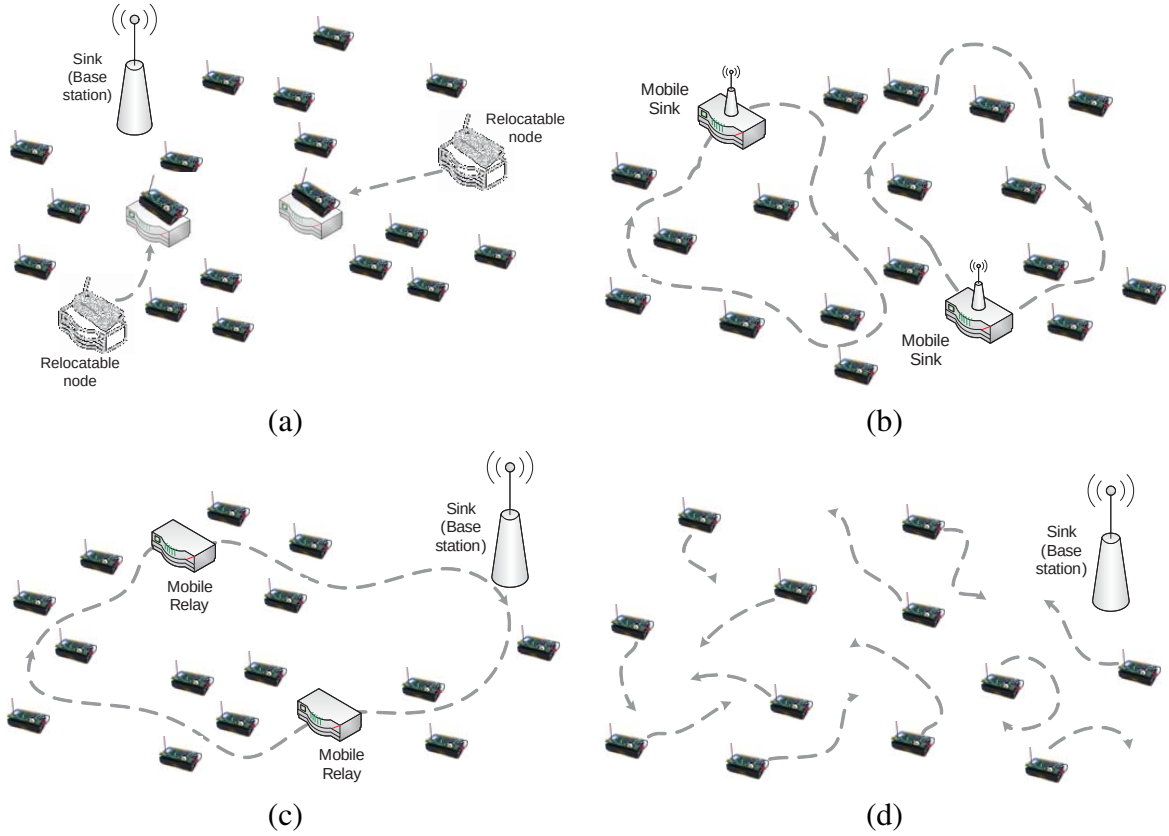


Figure 2.1: Different types of mobile nodes in a low-power network; (a) relocatables (data generator nodes), (b) sinks (data collector nodes), (c) relays (data collector nodes), and (d) peers (data generator or data collector nodes) [DFDA11].

Mobile relays. In other cases, mobile sinks are moving arbitrarily (Figure 2.1(c)). In urban scenarios, people act as mobile sinks by collecting environmental data such as weather or traffic conditions for a community of interested parties [ABCG11]. Special nodes may also act as mobile nodes that carry and relay data to the sink nodes. Unlike the sink nodes, relay nodes are not endpoints that collect data, but only act as forwarders, also known as "data mules" [SRJB03]. In this context, the source nodes are static and wait for a mule to be within the communication range before starting data transmission. Then the mule collects data and moves to a different location for collecting more data or for relaying the data to the sink node.

Mobile peers. These nodes are ordinary mobile nodes that can act as data generators and/or relays. When a peer node is within the communication range of a sink, it transfers its own data and the data that were collected and stored from other neighbor nodes. Fig-

ure 2.1(d) illustrates the architecture of a network with mobile peer nodes. These nodes are used e.g. in wildlife monitoring applications, such as tracking zebras in the ZebraNet project [JOW+02] or whales in the SWIM system [HS06]. Sensor nodes are attached to the body of an animal acting as peer nodes. They gather their own data and other static nodes' data, which are then transferred to a sink node.

In this Thesis, we assume that all nodes have the possibility to generate and collect data, while some nodes have the ability to move.

2.1.5 Mobility support at two protocol layers

In general, mobility can be tackled at either MAC sub-layer or at the Network layer [SSB14]. Below, we present a brief description of each strategy.

Mobility support at the MAC sub-layer. Node mobility causes deterioration of communication links, which is more severe in LPWNs. Once the link degradation is detected, one of the following strategies can be triggered [ZD10]: (i) the transmission continues if the mobile node is able to complete the transmission before link loss, (ii) the transmitter and receiver negotiate for a dynamic rate of data transmission that requires early transmission of data packets, and (iii) the transmitting node breaks the link and searches for a better link. These decisions require the Link layer to interact with the Application, Network and Physical layers. In LPWN, due to the resource constraints in terms of storage and data rate, the last strategy is more favorable.

Mobility support at the Network layer. Devising a mobility management strategy in the Network layer requires dealing with the following aspects: (i) *Node addressing:* some protocols assign unique IDs to nodes, such as Zigbee and WirelessHart; instead, 6LoWPAN and RPL are based on IPv6 addressing. (ii) *Low-power link features:* mobility support should respect the limitations of the network; in MIPv6, the restrictions of low-power networks are not considered. (iii) *Latency:* a mobility mechanism such as hand-off leads to a certain communication latency; in real-time applications, guaranteeing delay bounds is paramount. (iv) *Signaling cost:* the amount of message exchanges for handling mobility varies according to the mobility management mechanism; LPWNs require a minimum amount of signaling in order to prolong the network lifetime.

We tackle mobility at both MAC sub-layer (by devising a hand-off mechanism) and Network layer (by integrating the hand-off mechanism in a COTS routing protocol).

2.1.6 Mobility handlers

Mobility can be handled by different entities. As described before, there are various types of mobile entities such as source nodes, sink nodes and special nodes. The mobility handlers are classified as follows:

Node-based mobility support. The mobility is handled by regular sensor nodes by assuming some communication and application-specific tasks. In this way, the existing nodes within the network cooperate with each other in order to tackle mobility.

Network-based mobility support. In this approach, sensor nodes perform their regular functionality, while special nodes are used to handle mobility, implementing mobility mechanisms such as hand-off, re-addressing, re-routing and route optimization. By adding special nodes (with a more complex functionality) to the infrastructure, the regular sensor nodes are assigned lighter tasks.

Hybrid solution. This approach requires partial mobility support from the regular sensor nodes and partial support of additional nodes to the infrastructure.

We handle mobility by regular nodes, without the need of adding special nodes.

2.2 Data collection in a mobile network

In this section, we outline the most relevant tasks within mobility management. We start by defining the mobility detection, which is then followed by data transfer from the mobile node to the infrastructure and finally routing these data toward the sink node. Mobility features such as speed and trajectory affect all these phases. We briefly describe all these mobility features as follows.

2.2.1 Mobility detection

We define *mobility detection* (or *mobility discovery*) as the process that allows detecting a moving node. This process can take place at a static node, at the mobile node or at both of them. Since a reliable communication between the mobile node and the fixed node infrastructure is only possible upon good link quality, a good link should last long enough for data exchanges. The network should also be able to detect sudden and infrequent movements. To do so, there are two main approaches: (i) *pattern-independent mobility detection* schemes, and (ii) *pattern-dependent mobility detection* schemes. We describe these schemes below.

Pattern-independent mobility detection scheme. In this scheme, the mobility detection is independent of the mobility pattern. It is classified into three groups: *scheduled rendezvous*, *on-demand*, and *asynchronous*. In a *scheduled rendezvous* scheme, the static and mobile nodes agree on when (time instants) they meet each other. This requires a very precise timing schedule to know these exact moments. Nodes are sometimes synchronized through a global positioning system (GPS), such as in ZebraNet [JOW+02]. In an *on-demand* scheme, the static node wakes up based on an event that is initiated by the mobile node. Nodes can use multiple radios; a long-range and high-power radio for data communications and a low-range and low-power radio for wake-ups. The low-power radio continuously monitors the activities. The mobile node broadcasts control packets. When the mobile node detects activity on the radio channel, it turns on the high power radio for data communication [STGS02]. In an *asynchronous* scheme, nodes communicate without agreeing on the wake-up schedules. Nodes respect their periodic wake-ups without considering the mobility pattern. The mobile node periodically sends control messages, while the static node may receive them during its short listen period. If the static node detects the signal, it can start data transfer/receive, otherwise it turns back to sleep mode [STS02]. To ensure receiving the control messages or increasing the chance of reception, some parameters at the mobile node side and the static node side should be considered to tune the beaconing intervals and the duty cycling of the wake-ups [ACDF09].

Pattern-dependent mobility detection scheme. In this scheme, it is possible to exploit some knowledge of the mobility pattern of a node to get better efficiency. However,

even in a deterministic mobility pattern, the visit times between a mobile and a static node may be hard to predict. Some works propose a learning phase to check the presence of the mobile node [GZD11]. In this approach, the mobile node continuously emits beacons and the static node periodically listens. The static node stores the time when beacons are received, which is used for scheduling the wake-up intervals. For a random mobility pattern, static nodes try to learn the pattern, which is more challenging than the deterministic mobility. Perhaps, additional knowledge is required to tune the cycles. In [JAZ05], statistical information about the mean and the variance of visits (when there is a good link quality between the static and the mobile node) is exploited.

We consider an asynchronous pattern-independent mobility detection scheme.

2.2.2 Data transfer

We define *data transfer* as the process of single-hop message exchange between a mobile node and a static node. The data transfer process has to be aware of the mobility pattern, since the distance between the sender and receiver, speed of mobile node and the visit times affect the reliability of the link. **Experiments revealed that in low speed mobility, ranging from** 0.3 **to** 1.5 **m/s, the reliability of data transfer is independent of the speed** [KSJ+04b]. The situation with high-speed mobile nodes is different. Different speeds of the mobile node as well as different distances between the mobile and static nodes have been investigated [KSJ+04b]. The results showed that the probability of packet loss in high speed mobility (40 km/h or 11.1 m/s) is much higher than low speed mobility [KSJ+04b]. Moreover, the packet loss is minimum when the mobile node is close to the static node.

In addition to the mobility pattern, the mobility detection scheme also affects the process of data transfer. In some works, authors proposed a joint mobility detection and data transfer scheme. They derived a stability condition based on different parameters, considering the arrival rate of the mobile node and the message generation rate at the sensor node. They showed that the packet delivery ratio is $\approx 60\%$, when the stability

condition is satisfied [JSB+06]. However, in bursty arrival rates of the mobile node, the performance degrades.

In this Thesis, we consider low speed mobility and provide a mobility detection approach.

2.2.3 Route maintenance

The routing strategy and routing maintenance is the process of defining the message path between the mobile sources and the sink nodes. In case of a mobile sink, the routing strategy becomes more challenging. Note that in this Thesis, we assume static sinks. There are two main classes of routing techniques for uncontrollable mobile nodes; (i) *flat routing*, and (ii) *proxy-based routing*, which are identified below.

Mobile sink in flat routing. There are various approaches that consider all nodes have equal routing capabilities. In a modification of the optimized link state protocol (OLSR) [CJA+03], a proactive routing protocol for ad-hoc networks, the direction of the mobile node and the position of fixed nodes are used to enhance routing. The results showed that route changes decrease by 10% by considering the direction, and by 20% when exploiting both direction and positions [DS09].

Another example is the directed diffusion (DD) routing protocol that has been initially designed for static networks [IGE+03]. In this routing protocol, the sink propagates an interest, which is a query message referring to a specific event. Then this interest is disseminated throughout the network. Finally, multiple paths from sources to the sink are set up. To enable mobile sinks, two strategies were followed [KSJ+04a]. First, giving higher priority to the query message propagation coming from the sink and directly reaching the source node. This reduces redundant propagation to intermediate nodes. Second, exploiting acknowledgement messages during message exchanges, since the mobile sink moves and the messages may not have sufficient time to be successfully delivered.

The MintRoute protocol was designed specifically for the all-to-one data communication pattern [WTC03]. It was proposed for static networks, where data is routed in a tree-like fashion. The routing paths are evaluated by using a metric that minimizes the

number of retransmissions. A modification of MintRoute is MobiRoute [LPP+06], which features three main strategies to cope with mobile sinks: (i) using beacons and time-outs to detect broken links, (ii) limiting tree reconstruction to reduce overhead, and (iii) using bandwidth throttling and data buffering to mitigate data losses generated by mobile nodes.

Mobile sink in proxy-based routing. In this scheme, routing protocols use additional nodes called proxies (sometimes also referred to as gateways, access points, anchors or rendezvous points). When the mobile sink is in proximity of a proxy, the proxy starts transmitting buffered data toward the sink.

The two tier data dissemination (TTDD) scheme proactively builds an overlay grid-based forwarding structure to transfer packets to the mobile sink [LYC+05]. The intersection of grid lines are called crossing points, while squares are referred as cells. The source nodes transmit data to the four nodes on adjacent intersections. The location of these intersection nodes is advertised at the beginning of the experiment to reduce network overhead. The intersection nodes that are closer to the destination act as proxies. The mobile sink queries its local cell by sending a message. The intersection nodes query the message along the cell and collect the data to be routed toward the sink node.

In the Maximum Amount Shortest Path (MASP) protocol, the sensor nodes that are within a one-hop distance from the mobile sink are elected as proxies [GZD11]. These proxies collect data from the network and then transfer them to the sink. MASP assigns nodes as proxies with respect to some predefined constraints, aiming at keeping the energy consumption low and the throughput high. MASP runs in two phases: (i) to discover mobile sinks in the network, and (ii) to gather data at proxies.

There are several shortcomings in using proxy-based routing schemes. Most of these approaches are centralized. In case of having multiple mobile sinks, employing a centralized approach is more expensive as it requires many nodes with higher capabilities. In addition, many solutions assume that the location of nodes are known by using GPS devices. However, using GPS is costly, power hungry and cannot be used in many application environments.

In this Thesis, we consider static sinks.

2.2.4 Mobility pattern control

From the system perspective, node mobility is either controllable or uncontrollable. The mobility pattern can have a significant impact on the performance of data collection. In a controllable mobility pattern, an accurate mobility pattern can optimize network performance. It also gives more flexibility to the network designer. In this context, many robotized networks have been developed, such as; Robomote [DRS+05] and PackBot system [oT14]. Technically, a mobility pattern is characterized by two parameters: (i) trajectory, and (ii) speed, as outlined next.

Nodes Trajectory. The path followed by a mobile node is called trajectory. In a controlled mobility pattern, trajectory is controlled by employing either a static or dynamic scheme. If the path just changes over time, it is considered a static trajectory, and if the path changes on-the-fly, it is a dynamic trajectory. Many solutions have addressed the static trajectory control. For instance, in [LH05], a single mobile node collects data from a circular wireless sensor network (WSN). They showed that using a shortest path routing strategy implies moving at the border of the sensing area. The dynamic trajectory mobility pattern is classified into *on-demand* and *priority-based* approaches. In the on-demand scheme, the trajectory changes when an event is detected by the static node. This scheme is used in [ZAZ04] in such a way that the mobile node changes its trajectory upon receiving a request from a static node and deviating its path toward it. In the priority-based scheme, the trajectory change depends on the constraints (buffer overflow and latency). In [GBE06], an urgent message has the top priority and has to be collected within a specific time bound. These messages are usually generated infrequently with respect to the ordinary messages.

Node Speed. In a controlled mobility pattern, the mobile node can stop for data transfer. Depending on the path and the message generation rate at the source nodes, the stopping duration can change. This will in turn affect the mobile node speed. The Stop to Collect Data (SCD) algorithm has a similar approach [KSJ+04b]. The speed of the mobile node is tightly related to the maximum latency allowed by the application. With higher node speed, the stopping duration can be extended. The speed is changed only according to the number of nodes with data to transmit.

We assume that the mobility pattern of mobile nodes is uncontrollable and unknown.

2.3 Hand-off in wireless networks

2.3.1 Main concepts and definitions

A lot of research has been done on hand-offs in cellular networks and WLANs. In general there are two types of hand-offs: (i) *horizontal hand-off*, and (ii) *vertical hand-off*, as briefly explained next.

Horizontal hand-off. It occurs when a mobile node moves out of the coverage range of an AP into the coverage of another AP within the same wireless technology. For instance, if the node is connected with a 802.15.4 radio, a horizontal hand-off must lead to a new connection to another 802.15.4 radio. This concept is similar for cellular and WiFi networks.

Vertical hand-off. It is defined as the hand-off between two APs that use different wireless networking technologies, e.g., WiFi to and from cellular wireless networks.

In this Thesis, we address a horizontal hand-off technique for LPWNs using IEEE 802.15.4 radios.

Generally, a hand-off process has three main phases: (i) the *measurement phase*, where the results of this phase are the measurements from the physical environment, (ii) the *hand-off decision*, which is usually performed by hand-off algorithms based on some parameters and optimization criteria, and (iii) the *execution of hand-off*, via packet exchanges and/or changing radio channels or routing tables.

Hand-off may affect the quality of service in a wireless network. Hence, there are few desirable features and requirements that should be taken into account when designing a hand-off mechanism:

– *Hand-off delay.* The hand-off delay should be as small as possible, i.e. hand-off should be fast enough so that the network disconnection becomes minimal. This increases the chances of successful packet delivery.

– *Ping-pong effect.* The hand-off process requires extra packet exchange and delay, so unnecessary hand-offs degrade the network performance and should be avoided.

– *Packet delivery ratio.* An efficient hand-off should guarantee an acceptable packet delivery ratio by assigning good quality links between the mobile nodes and the fixed network infrastructure.

– *Network overhead.* Efficient and fast hand-off process should have minimum impact on the overall network overhead in terms of number of control packet exchanges.

There are some general factors that affect the hand-off process. When designing a hand-off mechanism, the following aspects should be considered:

– *Radio technology.* Each radio technology has its own features and limitations. For instance, in cellular networks, the cell size, cell topology and traffic models vary for different technologies, such as Global System for Mobile Communications (GSM), universal mobile telecommunication system (UMTS) and long term evolution (LTE). However, in LPWNs, e.g. in IEEE 802.15.4, a hand-off mechanism should deal with limitations such as single-radio nodes and unreliable links.

– *Network topology.* The network topology is one of the basic characteristics impacting the hand-off mechanism. The existence of a fixed infrastructure of static base stations helps in devising the hand-off mechanism.

– *Mobility model.* The speed and direction of the mobile node affect the hand-off performance. A fast movement requires faster hand-off decisions.

– *Quality of service (QoS).* Each application has its specific QoS requirements. For instance, in time-critical applications, it is paramount to perform fast hand-offs for delivering high priority messages.

2.3.2 Examples of hand-off mechanisms

Hand-off has been widely studied in cellular networks [CJC09, WL97, MHT00, SF06, MDT08] and wireless local area networks [RS05, MSA03, PJKC05, SMA04], but it has

not received the same level of attention in LPWNs. In this subsection, we briefly describe some of the most relevant hand-off schemes.

Hand-offs in cellular networks. In cellular networks, the hand-off decision is centralized and typically coordinated by a powerful base-station, which is able to leverage considerable information about the network topology and client proximity [CJC09]. Cellular networks also take advantage of sophisticated code division multiple access (CDMA) radios to perform soft hand-off techniques [WL97]. The major challenge in cellular networks is the *call dropping* effect during an ongoing call while switching between base-stations [MHT00]. A similar event occurs due to the lack of available channel, so-called *call blocking*. In [MDT08], some channels are exclusively allocated to hand-off calls, also known as *guard channels*. In [SF06], a queuing strategy has been applied to delay the hand-off calls until a channel becomes available. The hand-off strategies used in cellular network technologies with resourceful devices and multiple radios are not suitable for LPWNs with resource limitations.

Hand-offs in WLANs. Contrary to cellular systems, WLANs have a distributed architecture, where mobile nodes have no a-priori knowledge of the local network [RS05, MSA03]. While cellular systems require a continuous monitoring of the radio signal quality, WiFi-based systems monitor signals only after service degradation. The main concern of 802.11 hand-off protocols is to minimize the hand-off latency. A hand-off in WiFi-based systems is divided into the *Discovery* and *Re-authentication* phases. The channel scanning during the Discovery Phase is the most time consuming process. The authors in [PJKC05] propose a MAC sub-layer solution with fast hand-off mechanism that uses selective scanning and records the scan results in the AP's cache. When a mobile node (MN) moves to a location visited before, it pings the nearby access points (APs) for their available channels. In [SMA04], each AP records the neighboring AP's information in a *neighbor graph* data structure. Then the AP can inform the MN about which channels have neighbor APs and the MN needs to scan only those channels.

The key difference between WiFi and LPWN hand-offs is that in WiFi multiple radios are used to reduce the hand-off latency, while in LPWN a single radio is typically used. In LPWNs, a centralized hand-off approach is not feasible as it incurs in a high overhead.

Hand-offs in LPWNs should be distributed — similar to WiFi networks — while using a single-channel radio that focuses on the up-link and that can cope with the high variability of low-power links.

Hand-offs in LPWNs. There are two major strategies to design a hand-off mechanism: soft hand-off and hard hand-off. In this Thesis, we will address the design of these hand-offs: in Chapters 3 (hard hand-off) and in Chapter 7 (soft hand-off). Below, we provide few hand-off examples in LPWNs. Chapter 5 (subsection 5.3) provides detailed description on the mobility support using hand-off approach in 6LoWPAN and RPL routing.

In [YKH14], the authors propose a soft hand-off scheme by using the information of neighbor APs. The MN gathers beacon information (IDs, channel number and link quality (LQ) value) from neighbor APs. Each MN broadcasts a message containing its collected information in a designated time-slot during the Contention Free Period (CFP). Variation of the LQ shows the relative distance of MN to an AP. When the LQ with the serving AP drops below a predefined hand-off threshold, MN triggers re-association to one of the available APs in the neighborhood. The results indicated that the hand-off delay and the packet loss are very low. However, this scheme was not evaluated with a network simulator and real experiments.

In [PK11], the problem related to the mobility of sensor nodes (SNs) to hand-off between different gateways (GW) connected to the backbone network is addressed. It proposes a soft hand-off decision for WSNs based on 6LoWPAN (SH-WSN6), which avoids unnecessary hand-offs when there are multiple GWs in the range of SNs. The sensor node is able to register to multiple GWs at the same time by using IP solution. The SH-WSN6 takes advantage of router advertisement (RA) message defined in the Internet Control Message Protocol (ICMP). GWs transmit RA messages periodically to advertise their presence. At first, SN can register to only one GW. By receiving RA in each interval, the SN decides for the best GW. Every time an SN registers with a new GW, it gains a new route. This improves connectivity by having route diversity. If there is an unreliable link, comparison algorithm makes a decision to remove that link and therefore improves the QoS since poor links will not be used anymore.

In [ZV11] two additional control messages are transmitted in order to support the

attachment of the MN to a new point of attachment. These messages are the Join and the Join Ack that are sent/received when the MN is still attached to the previous tree position. Therefore, the role of the dynamic topology control in soft-hand-off mobility is to support the re-attachment of the MN to a different tree position as a result of movement inside the test-bed area. In the hand-off decision rules some parameters are defined which are (i) RSSI threshold, (ii) better RSSI, (iii) number of lost packets, and (iv) packet loss percentage. These values are set according to the application requirements.

In [CLBR10b] authors address a hard hand-off mechanism and describe a wireless clinical monitoring system collecting the vital signs of patients. In this study, the mobile node connects to a fixed AP by listening to beacons periodically broadcasted by all APs. The node connects to the AP with the highest RSSI. The scheme is simple and reliable for low traffic data rates. However, there is a high utilization of bandwidth due to periodic broadcasts and hand-offs are passively performed whenever the mobile node cannot deliver data packets.

2.4 Conclusion

In this chapter, we defined our system model and some fundamental concepts in mobile wireless networks. We first outlined the types of mobility, mobility patterns and mobility models by stressing our specific assumptions: we consider that only source nodes can be mobile and follow a pedestrian mobility pattern, which in simulations we assumed the Random Walk mobility model. All nodes in our system model have the possibility of generating and collecting data.

We described mobility solutions at the MAC sub-layer and Network layer. We stated that our work will tackle both protocol layers by devising a hand-off mechanism and integrating it within a COTS/SOTA routing protocol. We presented different aspects of data collection in mobile networks, which are known as mobility detection, data transfer and route maintenance. Various types of hand-off mechanisms (horizontal and vertical) were described, which were followed by the identification of the main performance metrics for

evaluating hand-off mechanisms. We then provided some examples on hand-off mechanisms in cellular and WiFi networks. In Chapter 3, we address the design and preliminary evaluation of the smart-HOP hand-off mechanism.

Chapter 3

The smart-HOP hand-off mechanism

This section explains the basics of the proposed hard hand-off procedure (dubbed smart-HOP) for LPWNs. Considering the limitations of low-power networks and the main features of unreliable links, we address the parameters involved in the hand-off process, which is then followed by preliminary tests to verify the effectiveness of smart-HOP in a controlled environment. Finally, we study the impact of radio interference on the smart-HOP performance. The contribution in this chapter is mainly supported by publication [FZA+12].

3.1 Basics on smart-HOP

In this section, we provide the overall idea of smart-HOP and highlight the importance of three parameters involved in the hand-off design: *link monitoring frequency*, *hysteresis thresholds* and *stability monitoring*, and then we describe how to fine-tune these parameters in a controlled environment.

The smart-HOP algorithm has two main phases: *(i) Data Transmission Phase* and *(ii) Discovery Phase*. A time-line of the algorithm is depicted in Figure 3.1. To explain the algorithm, we assume that initially the mobile node (MN) is not attached to any access point (AP). This state is similar to the case when the MN disconnects from one AP and searches for a better AP. In both cases, the MN performs a Discovery Phase by sending *n* request packets (beacons) in a given window and receiving a *reply* packet from each

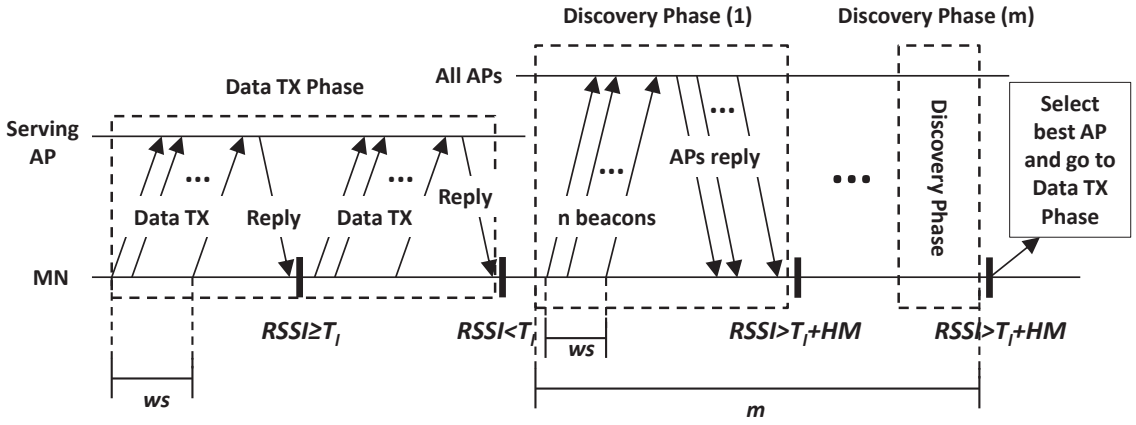


Figure 3.1: Timing diagram of the smart-HOP mechanism, illustrating the two phases and the messages exchanged.

neighboring AP. The reply packet holds the link quality level information that is defined as the average received signal strength (ARSSI), or average signal-to-noise ratio (ASNR), of the n packets received. By embedding the link quality level in the reply packet, the MN gets the down-link information and filters out the asymmetric links. Upon detecting a good link, the MN resumes a Data Transmission Phase with the AP offering the most reliable link. The data packets are sent in burst and a reply is received afterwards (similar to the Discovery Phase). This process enables monitoring the current link quality during the normal data communication process. The details of both phases are shown in Figure 3.1. The smart-HOP mechanism relies on three main parameters as follows:

Parameter 1: link monitoring frequency. It is an important parameter for any hand-off mechanism, which determines how frequent the link monitoring should be. The link monitoring property is materialized by the window size parameter (ws), which represents the number of packets required to estimate the link quality over a specific time. Considering an inter-packet interval of 10 ms with $ws = 3$, results in link monitoring frequency of 33 Hz. A small ws (high sampling frequency) provides detailed information about the link, but increases the processing (of reply packets), which leads to higher energy consumption and lower delivery rates. The packet delivery rate drops as the MN opts for unnecessary hand-offs, which are triggered by detecting low quality links due to sudden fluctuations of signal strength. On the other hand, a large ws (low sampling frequency) provides coarse grained information about the link and decreases the responsiveness of the network.

Parameter 2: threshold levels and hysteresis margin. The mobile node starts the *Discovery Phase* when the link quality goes below a certain threshold (T_ℓ) and looks for APs that are above a reliable threshold ($T_h \stackrel{\text{def}}{=} T_\ell + HM$, where HM is the *hysteresis margin*). The optimal values for the lower threshold level and the *hysteresis margin* are defined through experimental tests, as we are going to show in the next section. During the *Discovery Phase*, the mobile node sends *ws* beacons periodically, and the neighboring APs reply with the average RSSI or SNR of the received beacons. If one or more APs are above T_h, the mobile node connects to the AP with the highest link quality and resumes data communication; else, it continues broadcasting beacons (in bursts) until discovering a suitable AP. In order to avoid collisions, the APs use a simple TDMA MAC[1].

In LPWNs, the selection of thresholds and hysteresis margins is dictated by the characteristics of the transitional region and the variability of the wireless links. The lowest threshold is influenced by the boundaries of the transitional region. Studies have revealed that in real deployments of LPWNs, a large number of links ($> 50\%$) are in the transitional region [ZK04]. The exact threshold level within the transitional region is computed from simulation and experimental analysis. If threshold T_ℓ is too high, the node is likely to perform unnecessary hand-offs (by being too selective). If T_ℓ is too low, the node may use unreliable links. The HM parameter plays a central role in coping with the variability of low-power wireless links. If the hysteresis margin is too narrow, the mobile node may end up performing unnecessary and frequent hand-offs between two APs (ping-pong effect), as illustrated in Figure 1.4(a). If the hysteresis margin is too large, the hand-off may take too long, which ends up increasing the network inaccessibility time (and thus message delay) and decreasing the packet delivery rate.

Parameter 3: AP stability monitoring. Due to the high variability of wireless links, the mobile node may detect an AP that is momentarily above T_h, but the link quality may decrease shortly after being selected. In order to avoid this, it is important to assess the stability of the candidate AP. After detecting an AP above T_h, the mobile node sends m further bursts of beacons to validate the stability of that AP. The burst of beacons stands

[1] smart-HOP has been designed in a protocol-agnostic way. The idea is to be able to integrate smart-HOP in any MAC/routing protocol, with no or minimum modifications. However, we adopted a time division multiple access (TDMA) approach for smart-HOP message exchanges between the MN and the APs.

for the *ws* request beacons followed by the reply packets received from neighboring APs. Stability monitoring is tightly coupled to the hysteresis margin. A wide hysteresis margin requires a lower m, and vice versa.

Architectural design. smart-HOP has some distinct design features. Most hand-off methods in high-power wireless networks perform explicit disconnections, i.e., the node informs the old AP that it no longer needs it. smart-HOP does not perform these disconnections for two reasons. First, nodes' deployment may have limited overlap between neighboring APs — due to low coverage radios and low node density — and this limited overlap may not permit complex transactions (by the time a mobile node wants to disconnect, the AP may already be out of range). Second, removing explicit disconnections reduces the computational and transmission costs of mobile nodes.

The lack of explicit disconnections implies that the fixed infrastructure is not responsible to track the connectivity of mobile nodes (as opposed to what happens in e.g. cellular networks). Hence, the mobile node should take an active role in avoiding disconnections. This is simply done by maintaining a disconnection time-out. If the mobile node does not receive *reply* packets for a certain period of time, it starts the Discovery Phase. The time-out parameter depends on the timing requirements of the application; in our case it was set to 100 ms.

3.2 Why smart-HOP for LPWNs?

Hand-offs are used in all mobile networks. This simple concept of switching from one AP to another requires very careful design considerations, so that the application requirements and system limitations are respected. In this section, first we outline the main features of hand-off mechanisms in cellular and WiFi networks, and then by comparing the communication cost of these hand-off approaches, we show that our smart-HOP approach is more appropriate for low-power wireless networks.

In cellular networks, the base stations have high energy, processing, and communication resources. All the APs are connected through a stable wired backbone, which is

responsible for taking hand-off decisions. All mobile nodes periodically broadcast beacons along with their data packets. At the same time, the base stations communicate with each other and assess the location and the link quality of all mobile nodes. By detecting a low quality link, the base stations decide for the next serving base station.

In WiFi networks, power and bandwidth are more limited than in cellular networks. Thus, performing a centralized decision at the base stations (similar to cellular networks) is not efficient. In these networks, a distributed hand-off decision is performed at the MNs. All APs periodically broadcast beacons in various available channels with a precise timing (to eliminate overlapping). The MN periodically broadcasts request packets in all channels to get immediate replies (beacons) from neighbor APs. During the *Data Transmission Phase*, the MN gets periodic beacons from the serving AP. By detecting a low quality link from the serving AP and high quality link from one of the neighbors, the MN triggers the hand-off process.

smart-HOP can reduce the communication overhead. Applying the aforementioned techniques in LPWNs requires high amounts of beaconing, which in turn increases the network overhead, collisions and energy consumption. In low-power low-cost wireless networks with a limited backbone of APs, a centralized approach is not feasible. Moreover, periodic beaconing of APs in a single radio network leads to packet collisions. smart-HOP is a distributed hand-off mechanism where MNs are responsible for broadcasting beacons upon detecting a low quality link.

Let us assume a simple terminology to identify the communication overhead of smart-HOP. Denoting c_{tx}, c_{rx}, c_b, and n_{AP} as the transmission cost, reception cost, beaconing cost and average number of APs available[2]. The communication overhead of all wireless networks is formulated as follows.

1. in LPWNs with smart-HOP is $(c_{tx}+c_{rx})(1+\frac{1}{ws})$,

2. in LPWNs with broadcast approach is $c_{tx}+c_{rx}$,

3. in WiFi networks is $(c_{tx}+c_{rx})+(n_{AP}\times c_b+n_{AP}\times c_{rx})$ and

[2]The beaconing (process of transmitting beacons) is defined separately in order to be distinguished from the data transmission (c_{tx}). However, the cost of transmitting a data packet and a beacon is assumed equal.

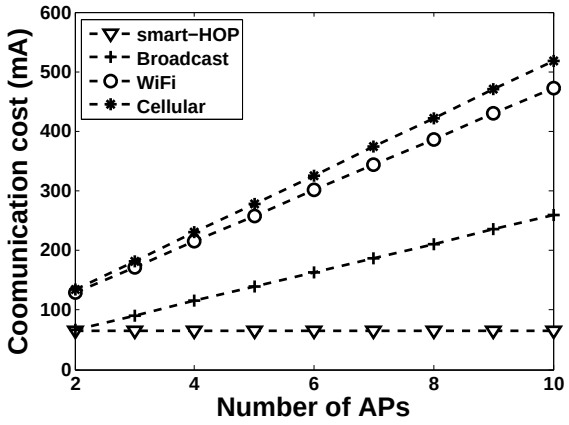


Figure 3.2: The communication costs of broadcast in LPWNs and the hand-off approaches in WiFi and cellular networks.

4. in cellular networks is $(c_{tx} + n_{AP} \times c_{rx}) + (c_b + n_{AP} \times c_{rx})$.

It is important to note that in estimating the costs, we considered simple hand-off approaches in each wireless technology. Simple manipulations lead to the following conditions[3].

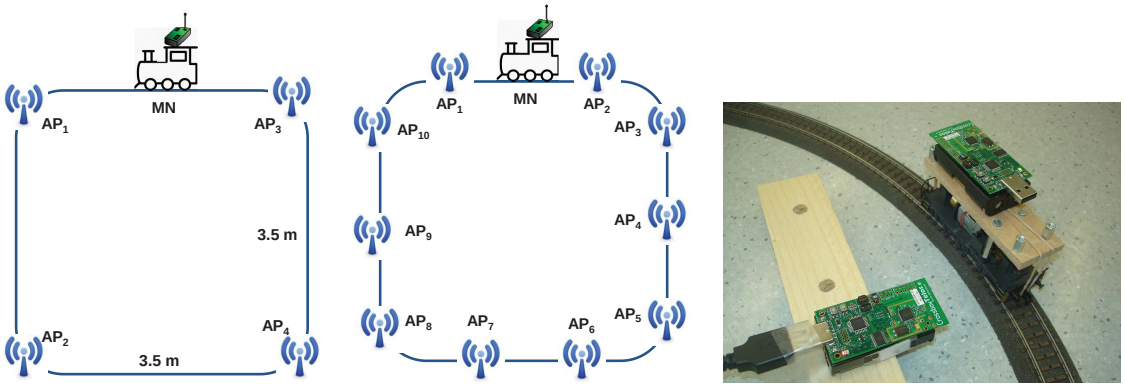
1. $\mathrm{C}_{\text{smart-HOP}} > \mathrm{C}_{\text{broadcast}}$ if $(ws \times n_{AP} - ws - 1)c_{rx} < c_{tx}$

2. $\mathrm{C}_{\text{smart-HOP}} > \mathrm{C}_{\text{cellular}}$ if $ws \times c_b + c_{rx}(2n_{AP} \times ws - ws - 1) < c_{tx}$

3. $\mathrm{C}_{\text{smart-HOP}} > \mathrm{C}_{\text{WiFi}}$ if $ws \times n_{AP} \times c_b + c_{rx}(n_{AP} \times ws - 1) < c_{tx}$

The only situation that verifies the above conditions is to have a very high transmission cost compared to the reception cost. In practice, transmission and reception costs for low-power radios such as the CC2420 radio are rather similar ($c_{tx} \cong c_b \cong 19$ mA and $c_{rx} \cong 24$ mA [CT14]). Hence, smart-HOP is expected to be more efficient than the broadcast approach in LPWNs and the conventional hand-off approaches in other wireless networks. The comparison between the cost of these processes is depicted in Figure 3.2.

[3]Two more conditions of $n_{AP} > 1$ (existence of more than one AP in the range of each MN) and $ws > 1$ (to apply a windowing process) are also respected.



(a) (b) (c)

Figure 3.3: The experimental set-up in the preliminary experiments: (a) 4 APs in the corners and a MN attached on top of a train, crossing APs, (b) 10 APs deployed on the boundary of the layout, and (c) a zoom-in view of the moment that MN passes by an AP.

3.3 Preliminary evaluation

3.3.1 Experimental set-up

We performed experiments in a controlled environment in order to fine-tune the smart-HOP parameters in a repeatable setting. A summary of the tests and major findings is presented here.

Calibrating the parameters of smart-HOP requires a test-bed that provides a significant degree of repeatability; a fair comparison of different parameters settings is only possible if all experiments are performed under similar channel conditions. In order to achieve this, we deployed a model-train in a large room. The room was 7 m$\times$7 m size and the locomotive followed a 3.5 m$\times$3.5 m square layout (Figure 3.3(a)). The speed of the locomotive was approximately 1 m/s (average walking speed). Figure 3.3(c) depicts a locomotive passing by an AP.

We implemented smart-HOP in TinyOS 2.0.2 and used TelosB motes for the evaluation. The smart-HOP implementation for the MN implementation required 19,754 Bytes of ROM and 936 Bytes of RAM and for the AP implementation required 18,720 Bytes of ROM and 810 Bytes of RAM. The inter-packet interval of the beacon and data packets was set to 10 ms. This value is close to the maximum rate possible, considering the processing, propagation and communication delays. The idea behind choosing the maximum

data rate was to evaluate smart-HOP close to its performance limits.

The first layout has four APs and one mobile node, as shown in Figure 3.3(a). Four APs were located at each corner of the deployment, and then increased to 10 APs, deployed on the boundary of the square layout (Figure 3.3(b)). The moment that the mobile node passes by an AP is depicted in Figure 3.3(c). To test smart-HOP under demanding conditions, we had to identify a transmission power that provides a minimum overlap among access points: $p_{out} = -25$ dBm satisfied this condition. Then, we ran several laps with the mobile node broadcasting packets. The experiments were performed at different times of day, during several days and with different number of people in the room. In all these scenarios, the mobile node required four hand-offs per lap; the time of day and number of people in the room (1 to 4 persons) do not seem to impact the number of hand-offs.

We utilized an interference-free channel to calibrate the parameters (Channel 26). The noise floor was constant (≈ -94 dBm). Our evaluation focuses on the impact that the hand-off parameters have on three network metrics:

Packet delivery ratio: the delivery rate of smart-HOP is compared to the best possible solution: broadcast. In a broadcast scenario, a packet sent by the mobile node can be received by any AP and there is no time spent on hand-offs.

Number of hand-offs: this metric captures the effectiveness in avoiding the ping-pong effect. The careful design of our test-bed guarantees a constant reference to evaluate this metric: four hand-offs per lap (the minimum number of hand-offs).

Mean hand-off delay: it represents the average time spent to perform the hand-off. Given that smart-HOP performs hard hand-offs, nodes cannot send data packets during this period. Hence, this delay should be minimized.

3.3.2 Parameter setting in experiments

The first step in a hand-off scheme is to determine when should a node deem a link as weak and start looking for another AP. In our framework, this is represented by T_ℓ. As already referred in Chapter 1, the *de-facto* way to classify links is to use the connected, transitional and disconnected regions. In order to identify these regions, we gathered RSSI and SNR

values at different parts of the building, utilizing different nodes. Figure 1.3 depicts these three regions for RSSI and SNR. An educated guess for the width of the hysteresis margin could be obtained from this figure (based on the 10 dBm width of the transitional region). However, while this value would guarantee that *all* links above T_h are reliable, it would also increase the amount of beacons and time required to reach T_h. In order to evaluate this region extensively, we consider different values for each hand-off parameter, as shown in Table 3.1. For example, if we consider scenario *A* with a 5 dBm margin and stability 2, it means that after the mobile node detects an AP above $T_h = -90$ dBm, the node will send two 3-beacon ($ws = 3$) bursts to observe if the link remains above T_h. Some studies have reported that three continuous beacon transmission provides accurate link-quality status for high sampling rates [AAH11]. The hysteresis margin *HM* captures the sensitivity to the ping-pong effect, and the number of bursts *m* captures the stability of the AP candidate (recall that each burst in *m* contains three beacons).

Table 3.1: Description of parameter setting for four different scenarios (thresholds, hysteresis margins, and AP stability monitoring).

Scenarios	T_ℓ	*HM*	*m*
A	-95 dBm	1, 5 dBm	1, 2, 3
B	-90 dBm	1, 5 dBm	1, 2, 3
C	-85 dBm	1, 5 dBm	1, 2, 3
D	-80 dBm	1, 5 dBm	1, 2, 3

We conduct experiments for all the scenarios as in Table 1. For each evaluation tuple $< T_\ell, HM, m >$, the mobile node performs four laps, resulting in a minimum of 16 hand-offs (4 per lap). The experiments provide some interesting results. First we show the results for the narrow margin (1 dBm), and then the ones for the wide margin (5 dBm).

3.3.3 Observations

The high variability of low-power links can cause severe ping-pong effect. Figure 3.4(a) depicts the total number of hand-offs for the narrow margin case. We observe two important trends. First, all scenarios are affected by ping-pong effect: the optimal number of hand-offs is 16, but all scenarios result in between 32 and 48 hand-offs. Due

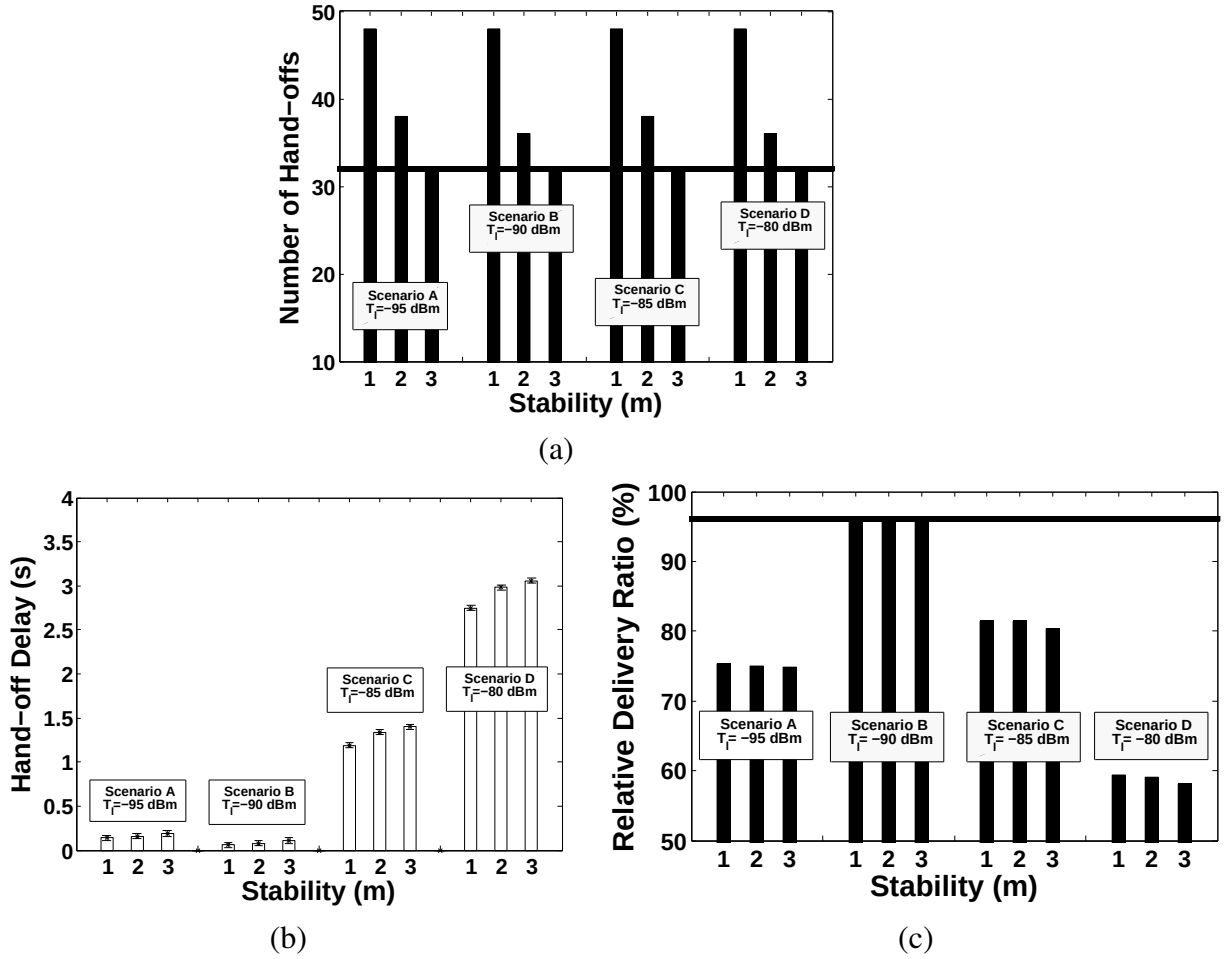


Figure 3.4: Results for narrow hysteresis margin ($HM = 1$ dBm). (a) number of hand-offs, (b) mean hand-off delay, (c) relative delivery ratio. The horizontal lines represent the results for the best scenario: 32 for the number of hand-offs and 96 for the relative delivery ratio. These values will be used as a reference in Figure 3.5 (horizontal line in (a) and (c)).

to the link variability, the transition between neighboring APs requires two to three hand-offs. Second, a longer stability monitoring m, helps alleviating the ping-pong effect: we observe that for all scenarios, the higher the m, the lower the number of hand-offs (Figure 3.4(a)).

Thresholds at the higher end of the transitional region lead to longer delays and lower delivery rates. Figure 3.4(b) depicts the average hand-off delay for various thresholds T_ℓ. A threshold selected at the higher end of the transitional region (-85 or -80 dBm: Scenarios C and D) can lead to an order of magnitude higher delay than a threshold at the lower end (-90 dBm: Scenario B). This happens because mobile nodes with higher

T_ℓ spend more time looking for overly reliable links (more time on Discovery Phase), and consequently less time transmitting data (lower delivery rate). Figure 3.4(c) depicts the relative delivery rate and captures this trend. In order to have a reference for the absolute delivery rate, we measured several broadcast scenarios considering a high transmission rate and a 4-access point deployment. We found that the average delivery rate was 98.2%, with a standard deviation of 8.7. This implies that there are limited segments with no coverage at all. Furthermore, the overlap is minimal, which tests the agility of the hand-off mechanism (as opposed to dense deployments, where very good links are abundant). Scenario A in Figure 3.4(c) is an exception, because the mobile node remains disconnected for some periods of time. No link goes below -95 dBm, hence, when this threshold is used, the mobile node will stay in Data Transmission Phase until the timer expires and resumes the Discovery Phase.

The most efficient hand-offs seem to occur for thresholds at the lower end of the transitional region ($T_\ell = -90$ dBm) and a hysteresis margin of 5 dBm. Figure 3.5 shows that Scenario B (-90 dBm) with stability 1 maximizes the three metrics of interest. It leads to the least number of hand-offs, with the lowest average delay and highest delivery rate. It is important to highlight the trends achieved by the wider hysteresis margin of HM=5 dBm. First, the ping-pong effect is eliminated in all scenarios of Figure 3.5(a). Second, contrarily to the narrower hysteresis margin (HM=1 dBm), monitoring the stability of the new AP for longer periods ($m = 2$ or 3) does not provide any further gains, because the wider margin copes with most of the link variability.

CSMA or TDMA. smart-HOP runs a simple TDMA MAC on the APs in order to avoid collisions and reduce the hand-off delay. Each AP performs a simple modulo operation on its unique *id* to obtain a specific time slot. For example, using 10 as the modulo operator, if a mobile node has neighboring access points with id's 16, 23, 45 and 72, the selected TDMA slots are 6, 3, 5 and 2, respectively. In theory, two nodes could collide, for example APs with *id*s 14 and 24 would select the same time slot 4, in practice, clock drifts and a relatively low density of access points (≤ 10) makes this unlikely. In our evaluation the modulo operator is 10 and the time slots are of length 5 ms (i.e., a TDMA

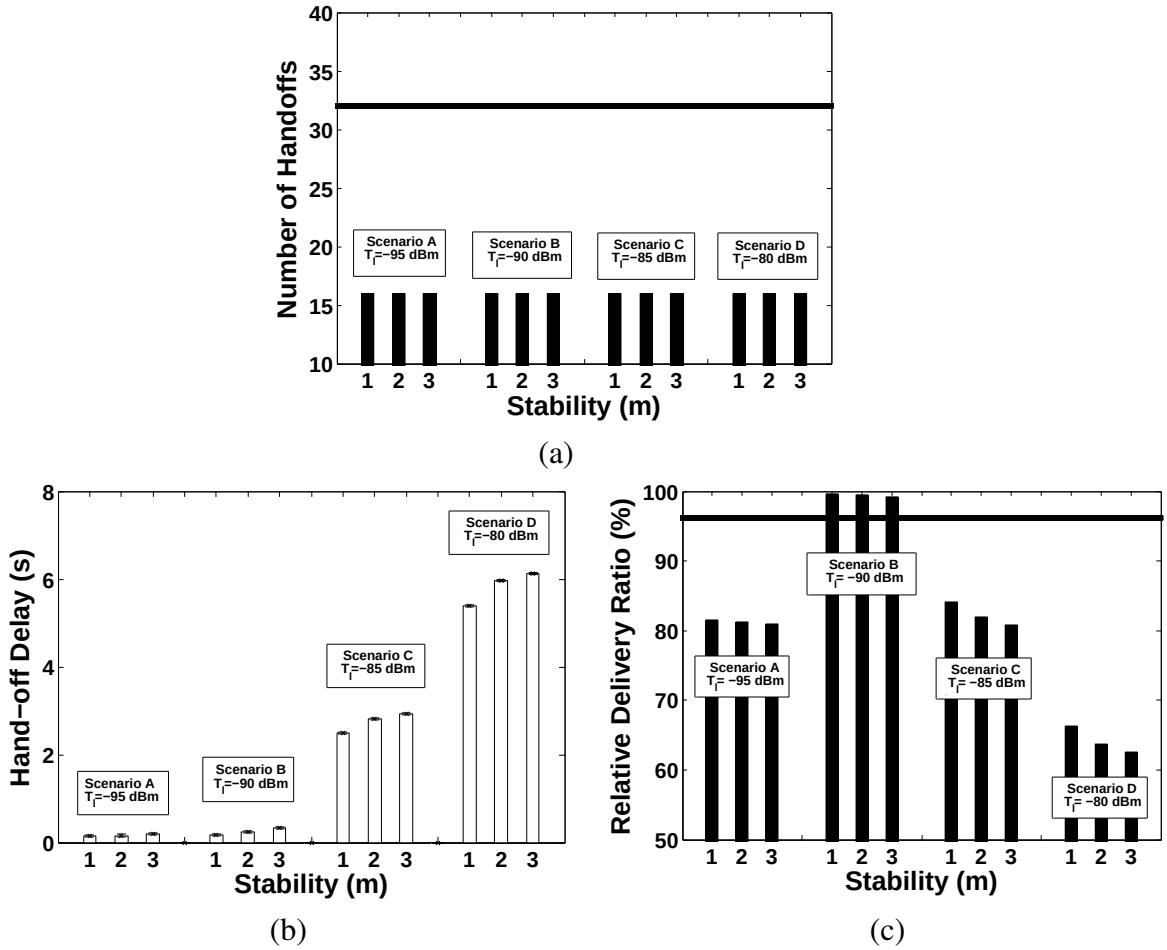


Figure 3.5: Results for wide hysteresis margin (HM=5 dB). (a) number of hand-offs, (b) mean hand-off delay, (c) relative delivery ratio. The horizontal lines represent the best results obtained for HM=1 (Figure 3.4). The lines highlight the importance of an accurate calibration of the hand-off parameters.

cycle of 50 ms)[4].

We compared the default carrier sense multiple access (CSMA/CA) MAC with the TDMA-based protocol for low AP density (4 APs — see Figure 3.3(a)) and high AP density (10 APs — see Figure 3.3(b)). Figure 3.6 shows that for Scenario B with 10 APs, the TDMA approach decreases the mean hand-off delay by half. For Scenarios C and D (results not shown), the type of MAC used has no impact, because the hand-off delays are very high (in the order of seconds).

[4]It is important to notice that this simple TDMA scheme is dynamic and can work on multi-hop networks. However, the density of APs should permit mobile nodes to have direct connectivity with at least one AP (i.e., single-hop communication)

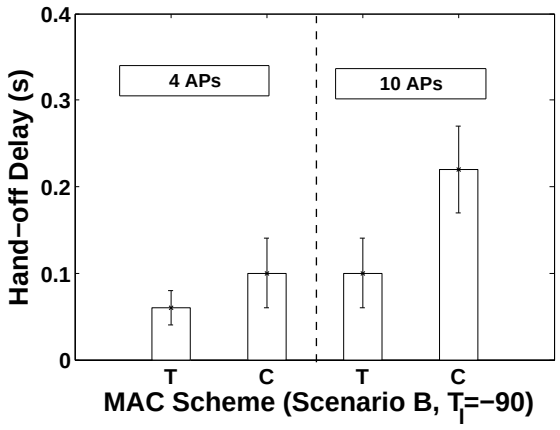


Figure 3.6: Mean hand-off delays for CSMA (represented as C) and TDMA (represented as T), with Scenario B ($T_\ell = -90$ dBm). TDMA always performs better.

3.4 Impact of interference

The performance of wireless networks is greatly affected by radio interference. Evaluating these effects is particularly important for LPWNs because they operate in the increasingly crowded unlicensed ISM bands. In more demanding applications (requiring certain levels of QoS), where information should be transferred in a timely and reliable way, it is a must to evaluate the impact of interference. We evaluated the performance of smart-HOP under different types of interference, and found that, unless the interference is bursty, strong and continuous (extreme adverse effects), smart-HOP copes well with interference.

When interference is likely to occur, smart-HOP should utilize SNR-based parameters instead of RSSI. This change of parameters is not a minor trade-off. Utilizing SNR implies that after receiving each packet, a node should sample the noise floor and subtract this value from the RSSI. Hence, if negligible interference is expected, RSSI should be the preferred metric for smart-HOP.

The tests are performed on Channel 15 of the CC2420 radio, which is affected by different sources of radio interference in the 2.4 GHz band such as WiFi devices and Bluetooth. In order to perform a systematic evaluation, we utilized the most common types of interference types/sources found in the ISM band as reported in [BVN$^+$11] as follows:

Periodic interference. This interference is usually spatially localized and has a regular

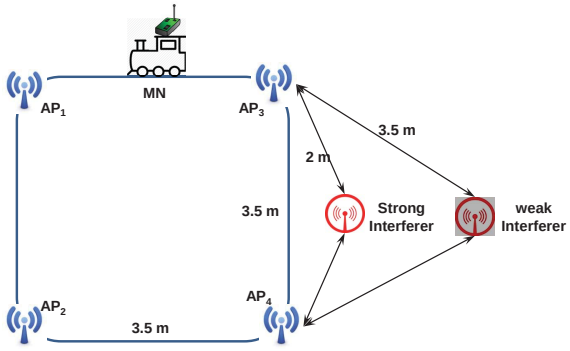


Figure 3.7: The layout with four APs and the interferer located 1 m and 3 m away from AP_3 and AP_4.

duty cycle. The best examples are microwave ovens. We recreate this interference pattern utilizing the Jamlab tool [BVN+11], which leverages COTS motes to generate customizable interference patterns. The power level of the interference is set to (-25 dBm, the minimum transmission power used for data communication), the period to 10 ms and the duty cycle to 50%. In our evaluation, we place the interferer mote at the right side of the deployment (Figure 3.7). This placement affects mainly AP_3 and AP_4. To test different interference levels, we place the interferer at 2 m from AP_3 and AP_4 (strong interference) and 3.5 m (weak interference). The noise pattern is shown in Figure 3.8(a).

Bursty interference. This interference is widespread and has a more irregular and random behavior. The best example is the interference caused by WiFi access points. To test this interference, we place a WiFi station inside the test room (strong interference) and on a different room six meters apart (weak interference). We have a laptop downloading a large file to maintain the interference for long periods of time. Figure 3.8(b) shows the interference observed at AP_4, but the interference is similar on all APs.

Figure 3.9 shows the performance of smart-HOP under interference. First, let us evaluate the results for the periodic interference scenario. The main observation is that, under periodic interference, smart-HOP with SNR leads to an increase on both the average hand-off delay and the delivery rate (right side of Figures 3.9(a) and 3.9(b)). When SNR is used, the affected APs (3 and 4) increase their delay by a factor of 4, but the overall outcome is positive, because the delivery rate increases by more than 10%. In the SNR case, a longer hand-off delay occurs, because mobile nodes spend more time in the Discovery

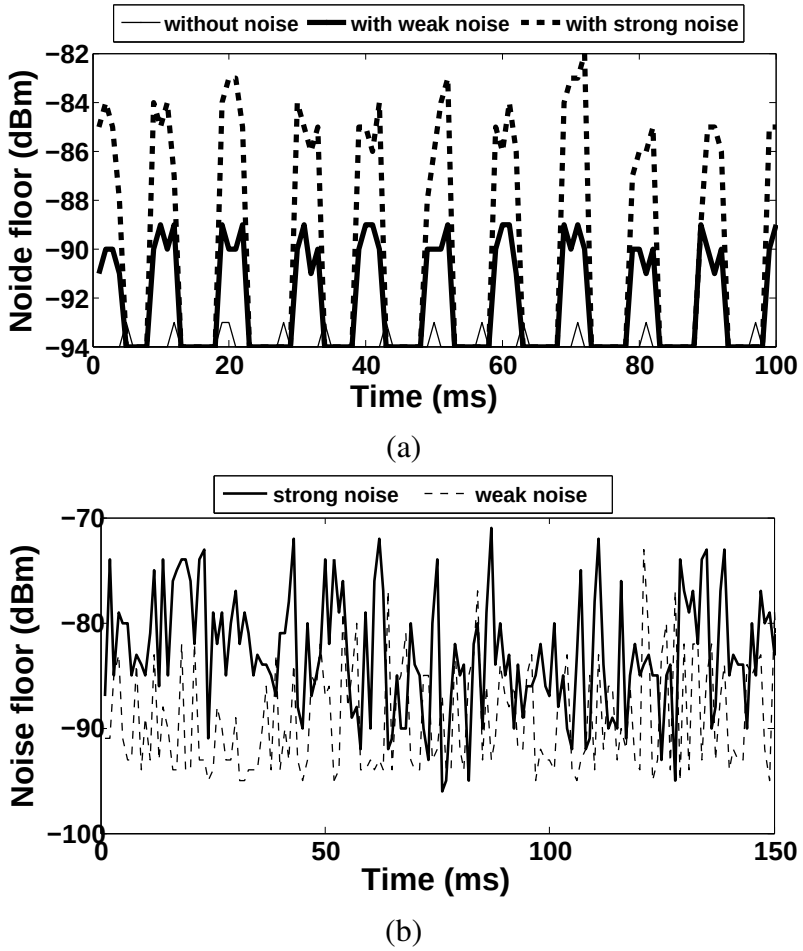


Figure 3.8: Noise floor for (a) periodic interference and (b) bursty interference

Phase looking for links that are good in spite of interference (SNR above T_h). On the other hand, RSSI-based smart-HOP may connect faster but to weaker APs (because RSSI alone cannot recognize the presence of interference). The higher delivery rate occurs because the SNR-based smart-HOP detects the presence of interference earlier and starts the Discovery Phase. On the other hand, RSSI-based hand-offs only react when packets are lost, and once they find a new AP, the link may not be strong enough for communication (again, because RSSI cannot detect interference by itself).

The bursty interference evaluation, Figures 3.9(c) and 3.9(d), highlights the effects of weak/strong bursty and continuous interference. With strong bursty interference, hand-off delay increases drastically and the packet delivery ratio drops by $\approx 50\%$. However, it should be mentioned that this is not a limitation of smart-HOP, but a fundamental limitation of low-power links. The broadcast scenario (an ideal one), does not perform much

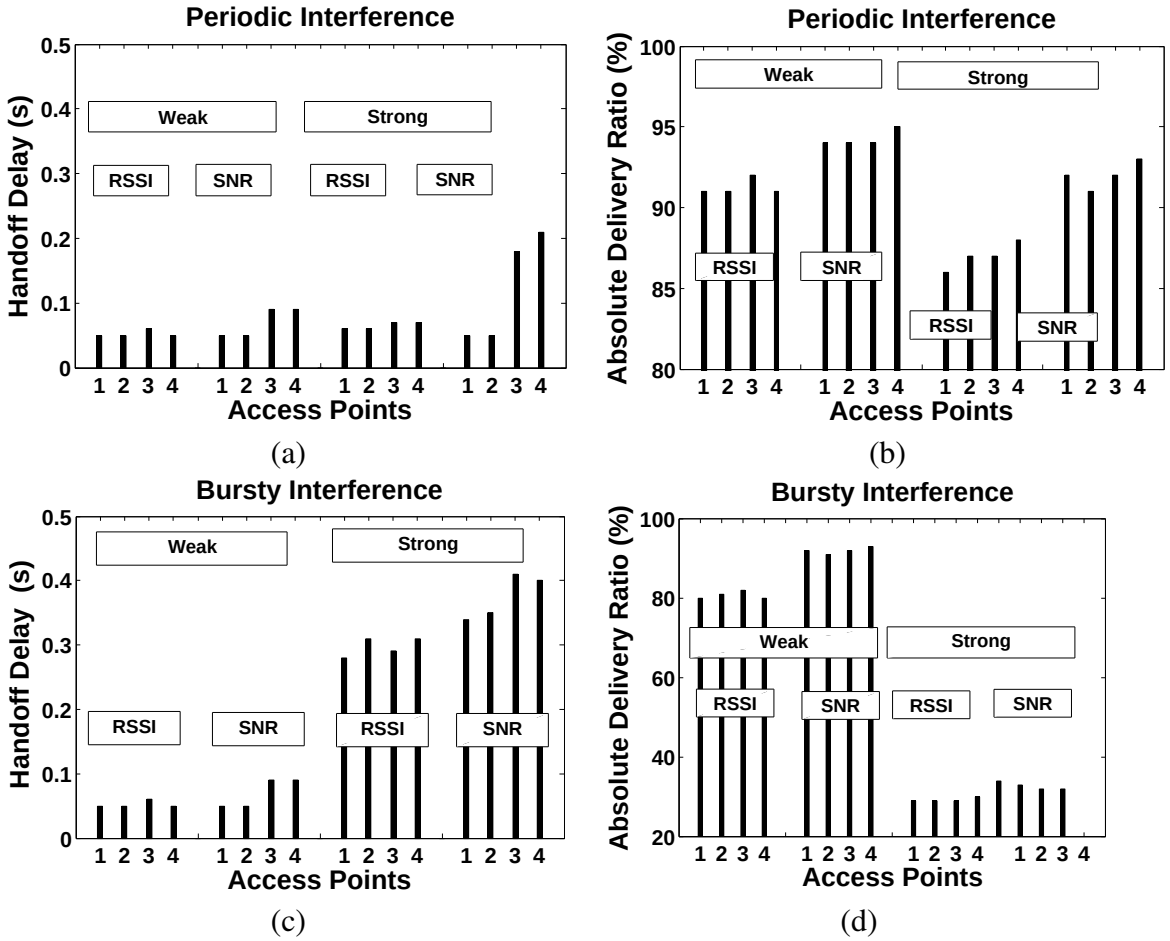


Figure 3.9: Effects of periodic and bursty interference at each access point. Periodic: (a) mean hand-off delay, (b) absolute delivery rate. Bursty: (c) mean hand-off delay, (d) absolute delivery ratio

better than smart-HOP. In fact, the *relative* delivery rate of smart-HOP with respect to the broadcast scenario is above 95%. In the presence of a continuous and strong interference, the possibility of successful packet delivery with a low-power radio is very low. Hence, the lack of information (data packets and beacons) enlarges the hand-off delay in both the RSSI- and SNR-based approaches.

3.5 Conclusion

In this chapter, we addressed the design, implementation and preliminary evaluation of a reliable and timely hand-off mechanism for LPWNs — smart-HOP. The hand-off mechanism is required for applications such as health-care monitoring where sensors collect

the vital signs of various patients. Any disconnection or long network inaccessibility times would have serious consequences. Indeed, the correctness of the application depends greatly on the proper management of mobility to ensure connectivity and real-time streaming. Therefore, the use of effective hand-off mechanism becomes paramount for such applications.

Compared to traditional cellular and WiFi systems, LPWN nodes performing hand-offs face two challenges: (i) their constrained resources limit the use of sophisticated hand-off techniques and (ii) their low-cost and low-power radio transceivers lead to highly variable wireless links, which affects the stability of the hand-off process.

We performed a carefully designed set of experiments under a controlled test-bed, based on IEEE 802.15.4 radios, to get a better insight on the settings of key parameters, namely, the lower link quality threshold level (required to start the hand-off, -90 dBm) and the hysteresis margin (required to finalize the hand-off and for stability, 5 dBm). We will further analyze and validate the smart-HOP mechanism analytically and empirically in the next chapter.

Chapter 4

smart-HOP evaluation

This chapter further evaluates the smart-HOP mechanism through analytical, simulation and experimental models. We study the impact of radio channel parameters (path-loss exponent and shadowing standard deviation) on the overall hand-off performance. We also elaborate on the impact of the link monitoring frequency and stability monitoring parameters with a simulation model. Then, we set-up a more realistic test-bed for a more precise fine-tuning the hand-off parameters and for further evaluating the hand-off performance. The contribution in this chapter is mainly supported by publication [FAZK14].

4.1 Analytical model and evaluation

The performance of hand-off in low-power wireless networks may greatly depend on the network layout and environmental conditions. This section addresses the impact of radio channel parameters on hand-off performance.

The two main radio channel parameters are defined as follows [ZK04]: (1) *Path-loss exponent (η).* It measures the power of radio frequency signals relative to distance. (2) *Shadowing standard deviation (σ).* It measures the standard deviation in RSSI measurements due to log-normal shadowing. The values of η and σ change with the frequency of operation, the clutter and disturbance in the environment. At this stage, we study the smart-HOP performance for various parameter settings, through a simple yet effective probabilistic model.

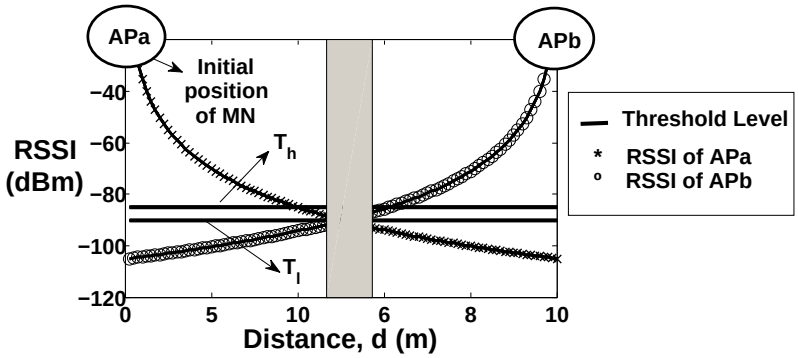


Figure 4.1: System model for the probabilistic analysis. It show the RSSI fluctuations of AP_a and AP_b in a 10 m distance. The threshold levels are assumed -85 and -90 dBm. The hand-off happens in the middle of this trip (shadowed bar) [FAZK14].

4.1.1 Probabilistic model

It is important to consider a radio propagation model that is reasonably faithful to the physical world model while still amenable to analysis. We assume a scenario consisting of two APs (AP_a and AP_b) and a MN. This assumption is sufficient, without loss of generality, as we are considering a hard hand-off process. In the probabilistic model, we assume the *link monitoring frequency* and *stability monitoring* $ws = m = 1$.

The two main hand-off performance metrics are the probability of ping-pong effect and the expected hand-off delay [YPMM01]. The system model and a general behavior of smart-HOP are shown in Figure 7.4. The two APs are separated by distance $d = 10$ m, while the MN moves from the vicinity of AP_a to the vicinity of AP_b along a straight line. In this model, the MN moves with a constant speed of 1 m/s. The received signal strength from AP_a declines till it reaches the lower threshold level T_ℓ, thus triggering the hand-off process. From this point onward, the MN stops communicating with AP_a and tracks the RSSI of the neighboring AP, AP_b. If the mobile node observes a signal strength above a higher threshold level, T_h, the hand-off process is considered to be finished. The hand-off duration (delay) is marked with a shadowed vertical bar in Figure 7.4.

Link quality monitoring. There are different ways of measuring the link quality metric. In this context, we consider RSSI as the link quality metric. The probabilities of RSSI being below the lower threshold level and above the higher threshold level are defined by a *Q-function* [KL07]. In this turn, the traveling path of the MN is divided into

a number of slots. For the sake of simplicity, we consider the same sampling rate for both the Discovery and Data Transmission Phases. These probabilities are expressed as follows.

$$P(R_a(i) < T_\ell) = Q(\frac{-T_\ell + R_a(i)}{\sigma}) \quad P(R_b(i) > T_h) = Q(\frac{T_h - R_b(i)}{\sigma})$$

Where $Q(.)$ is the complementary distribution function of the standard Gaussian, i.e., $Q(x) = \int_x^\infty (1/\sqrt{2\pi}) e^{-t^2/2} dt$, $R_a(i)$ and $R_b(i)$ indicate the RSSI values from AP_a and AP_b at slot i, and σ (in dB) expresses the standard deviation.

Radio channel model. The received signal strength is estimated by log-normal shadowing path-loss. According to this model, $R(i)$ (in dBm) (RSSI level at a given slot i) from the transmitter is given by [Rap96]:

$$R(i)[dBm] = P_t[dBm] - \overline{PL(d_0)}[dB] - 10\eta log_{10}(i/d_0) - X_\sigma[dB] \tag{4.1}$$

Where i corresponds to distance, P_t is the transmission power, $\overline{PL(d_0)}$ is the measured path-loss at reference distance d_0, η is the path-loss exponent, and $X_\sigma = N(0,\sigma)$ is a normal variable (in dB). The term X_σ models the path-loss variation across all locations at distance i from the source due to shadowing, a term that encompasses signal strength variations due to the characteristics of the environment (i.e., multi-path distortion, reflections, etc.).

smart-HOP probabilistic model. To evaluate the performance metrics, we define the probability of starting and ending the hand-off at each slot. Figure 7.5 shows an example where the MN encounters disconnections and connections in different time slots. As explained earlier, the MN communicates with a single AP at each time-slot. The MN is initially connected to AP_a — see Figure 7.4. When the MN is traveling from AP_a to AP_b, it tracks the likelihood of the RSSI going below a threshold level, T_ℓ, at each sampling interval. By observing a low quality link, the hand-off process starts. The probability of starting a hand-off at slot $s \in [1,k)$ is defined as follows (k indicates the total number of slots).

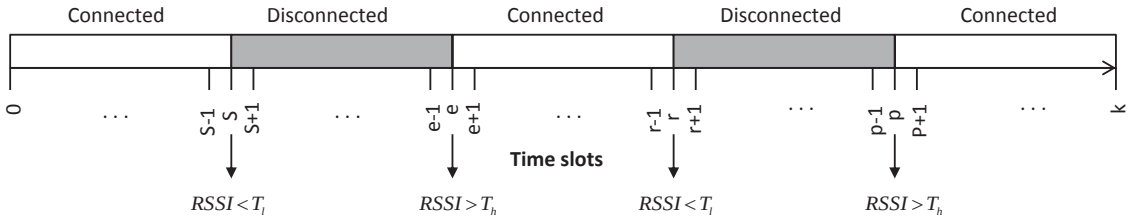


Figure 4.2: The moments of detecting different events; at slot $s-1$ the link quality level is below T_ℓ and starts a hand-off, at slot e is above T_h and stops the hand-off, at slot $r-1$ is again below T_ℓ, and at slot p is again above T_h. The shadowed areas are the disconnected regions during which the MN triggers the Discovery Phase [FAZK14].

$$P(S(s)) = \left[\prod_{i=1}^{s-1} P(R_a(i) \geq T_\ell)\right] \times P(R_a(s) < T_\ell) \tag{4.2}$$

The first part of the equation indicates the observation of a number of slots $(s-1)$ with good/acceptable link quality level (above T_ℓ). The second part denotes the observation of the low link quality for the first time (below T_ℓ). The following settings are used in all future evaluations across this section: $\sigma = 4$ dB, $\eta = 4$, $P_t = 0$ dBm, $d_0 = 1$ m, $d = 5$ m, $PL(d_0) = -55$ dB, $ws = m = 1$, $T_\ell = -90$ dBm and $T_h = -85$ dBm[1].

By starting the hand-off process at slot/location s, the MN disconnects from the corresponding AP that was servicing the MN. At this moment, MN starts assessing the other neighboring APs to choose the one with higher threshold level ($RSSI > T_h$). The hand-off finishes when the MN observes a high link quality. Equation 4.3 formulates the probability of ending a hand-off at slot e considering the fact that the hand-off would have been started at slot s and the MN was disconnected from either AP_a or AP_b.

$$P(E(e) \mid S(s)) = \left[\prod_{i=s+1}^{e-1} (P(R_a(i) < T_h) \times P(R_b(i) < T_h))\right] \times [1 - (P(R_a(e) < T_h) \times P(R_b(e) < T_h))] \tag{4.3}$$

[1]The values are set according to the following reasons: σ and η are selected according to an indoor environment values [ZK04], d_0, $PL(d_0$ and $P-t$ are the reference values in radio channel model [Rap96], ws and m are set to 1 to simplify the modeling and T_ℓ and T_h are set according to the most efficient scenario in the preliminary experiments.

This equation assumes that the hand-off occurs at slot s. The ending moment at slot $e \in (s+1,k]$ happens at a later stage by comparing the RSSI level of APs to a higher threshold level, T_h. Hence, in practice it is a conditional probability and depends on a situation that has taken place previously at slot s.

The time span between the starting slot and the ending slot is the hand-off delay. By considering each slot as a starting moment of a hand-off process, we characterize the ending moments by the probabilities defined in Equation 4.3. The expected hand-off delay is computed by getting the weighted sum of all possible hand-off periods. It is defined as the product of the time spent in each possible hand-off process started at slot s and ended at slot e by the correspondent probabilities of starting a hand-off at slot s, $P(S(s))$, and ending it at slot e, $P(E(e) \mid S(s))$. For each hand-off starting at slot s, the hand-off would end at one of the slots from $s+1$ to k. The sum of all these possible situations defines the expected delay for a hand-off started at a specific slot s. The overall expected hand-off delay is given by:

$$Delay(s,e) = \sum_{s=1}^{k-1} \sum_{e=s+1}^{k} ((e-s) \times P(E(e) \mid S(s)) \times P(S(s))) \tag{4.4}$$

In order to measure the ping-pong effect in smart-HOP, a new term is defined that is called probability of restarting a hand-off. This situation happens when a MN performs hand-off at an improper moment, thus leading to an unnecessary hand-off. The restarting of a hand-off always occurs after successfully ending the first hand-off at slot $r \in (2,k]$. This means that the probability of restarting is also a conditional probability that depends on ending a hand-off at an earlier stage. Since the MN may have been connected to either AP_a or AP_b, the signal strength should be evaluated for both cases. The equation is defined as follows.

$$\begin{aligned} P(R(r) \mid E(e)) = & \left[\prod_{i=e+1}^{r-1} (1 - P(R_a(i) < T_\ell) \times P(R_b(i) < T_\ell)) \right] \\ & \times [P(R_a(r) < T_\ell) \times P(R_b(r) < T_\ell)] \end{aligned} \tag{4.5}$$

The second disconnection of the MN in this trip will be ended at slot p, which is given by.

$$P(P(p) \mid R(r)) = \left[\prod_{i=r+1}^{p-1} (P(R_a(i) < T_h) \times P(R_b(i) < T_h)) \right] \times [1 - (P(R_a(p) < T_h) \times P(R_b(p) < T_h))] \quad (4.6)$$

To find out the probability of ping-pong effect, the full history of a MN since the first start of hand-off to the end and restarting again must be taken into account. Equation 4.7 illustrates the cases that lead to a ping-pong effect at slot p as follows.

$$P(P(p)) = \left[\left(\prod_{i=1}^{s-1} P(R_a(i) < T_\ell) \right) \times P(R_a(s) < T_\ell) \right] \times \left[\prod_{i=s+1}^{e-1} (P(R_a(i) < T_h) \times P(R_b(i) < T_h)) \right] \times [1 - P(R_a(e) < T_h) \times P(R_b(e) < T_h)] \times \left[\prod_{i=e+1}^{r-1} (1 - P(R_a(i) < T_\ell) \times P(R_b(i) < T_\ell)) \right] \times [P(R_a(r) < T_\ell) \times P(R_b(r) < T_\ell)] \times \left[\prod_{i=r+1}^{p-1} (P(R_a(i) < T_h) \times P(R_b(i) < T_h)) \right] \times [1 - P(R_a(p) < T_h) \times P(R_b(p) < T_h)] \quad (4.7)$$

To find out the total probability of ping-pong effect, we define the following equation (4.8) in a more abstract way.

$$P_{pp} = \sum_{s}^{k-3} \sum_{e}^{k-2} \sum_{r}^{k-1} \sum_{p}^{k} P(S(s)) \times P(E(e) \mid S(s)) \times P(R(r) \mid E(e)) \times P(P(p) \mid R(r)) \quad (4.8)$$

For each case of hand-off occurrence at slot $s \in (0, k-2]$, there is a chance to finish the hand-off at one of the upcoming slots $e \in (s, k-1]$. Similarly for each e, as an ending slot, there is a chance of restarting another hand-off at slot $r \in (e, k]$.

Performance metrics. To evaluate the performance of the smart-HOP mechanism, two main metrics are considered, which are derived from the above equations.

1. *probability of ping-pong effect.* It shows the probability of reconnecting to AP_a after the first hand-off from AP_a to AP_b. This situation happens after observing a low quality link $R_b(i) < T_\ell$ at AP_b at slot i, when there was a high quality link $R_b(i-1) > T_\ell$ at slot $i-1$.

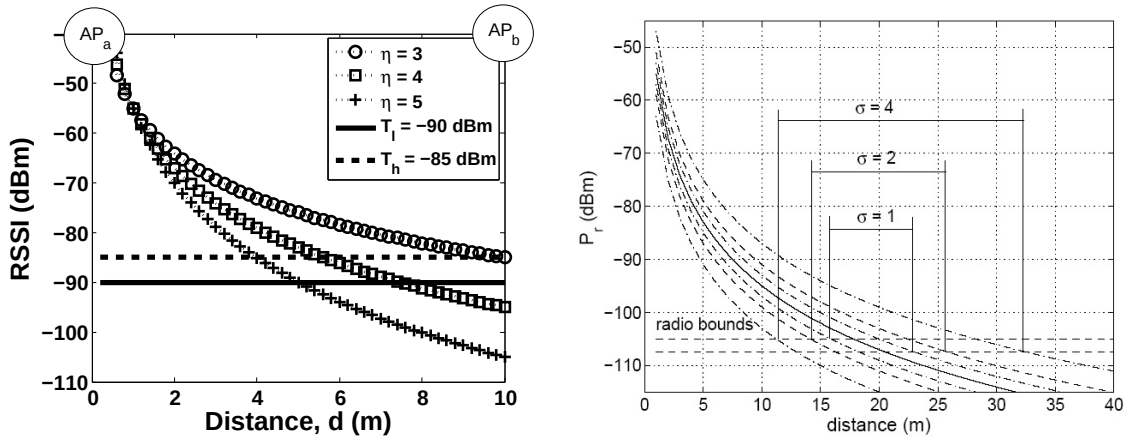
2. *expected hand-off delay.* It indicates the expected hand-off delay for each possible starting point of hand-off from AP_a to AP_b.

In the following subsections, we study the impact of some parameters that were either neglected or not feasible to consider in the preliminary experiments (Chapter 3).

4.1.2 Impact of channel parameters

An increase in the path-loss exponent (η) leads to longer hand-off delay and higher probability of ping-pong effect. The path-loss exponent varies depending on environmental conditions. The path-loss parameter may be less dynamic in some applications with stationary nodes and static environments or oppositely may be highly variable in some other situations like mobile LPWN applications [ABD10]. Figure 4.3(a) illustrates the variation of the RSSI at AP_a while the MN is moving toward AP_b. As expected, the larger the path-loss exponent is, the higher the slope of the RSSI decrease. In smart-HOP design, we aim at choosing a hand-off starting level below the intersection of the received signal power from AP_a and AP_b. This will reduce the chances of ping-pong effect. The ending level is supposed to be at this intersection, or slightly higher. In practice, it is recommended to pick a higher level for ending point to cancel out the sudden changes of the RSSI.

An increase in the shadowing variance (σ) enlarges the transitional region, which in turn causes higher link unreliability and ping-pong effect. This channel parameter describes the received signal strength fluctuation caused by flat fading. By increasing σ, the probability of entering the transitional region at closer distances from the transmitter and leaving it at farther distances increases; this results in a larger transitional region — see Figure 4.3(b) [ZK04].



(a) (b)

Figure 4.3: Impact of channel parameters on the RSSI with $T_\ell = -90$ dBm and HM=5 dBm, (a) impact of path-loss exponent on RSSI [FAZK14], and (b) impact of shadow fading on the RSSI (P_r indicates the received signal strength) [ZK04].

By enlarging the channel parameters (η and σ), the hand-off delay increases. Figure 4.4 depicts a matrix, with the x-axis representing the path-loss exponent and the y-axis representing the shadow-fading. We observe an increasing trend in the hand-off delay in each row and column when increasing each channel parameter. A larger path-loss exponent causes faster RSSI decrease. Hence, the MN enters the hand-off process at earlier stages. Finding a high link quality is postponed to later stages, which in turn increases the hand-off delay. Larger shadow-fading increases the RSSI standard deviation, which expands the transitional region. Therefore, any disconnection from the point of attachment requires a longer time assessing the wireless link to detect a high quality link in the transitional region.

Channel Parameters	η=1	η =2	η =3	η =4	η =5	η =6
σ=1	8.2e-099	1.4e-045	2.4e-014	0.0216	0.0387	0.0469
σ=2	3.1e-027	9.1e-014	2.5e-005	0.0324	0.0408	0.0470
σ=3	2.4e-013	2.8e-007	0.0022	0.0371	0.0422	0.0477
σ=4	2.6e-008	7.5e-005	0.0109	0.0398	0.0433	0.0496
σ=5	6.9e-006	0.0012	0.0215	0.0415	0.0443	0.0519
σ=6	1.6e-004	.0054	0.0297	0.0428	0.0453	0.0537

Figure 4.4: Impact of channel parameters on the overall expected hand-off delay (in seconds), sample rate=20 Hz [FAZK14].

Channel Parameters	η =1	η =2	η =3	η =4	η =5	η =6
σ=1	0	0	1.8e-229	6.4e-144	5.4e-079	2.1e-034
σ=2	5.7e-120	3.7e-087	5.3e-060	1.7e-038	6.6e-022	2.2e-010
σ=3	1.0e-054	4.2e-040	5.7e-028	2.7e-018	1.0e-010	2.4e-005
σ=4	1.6e-031	2.7e-023	1.7e-016	6.8e-011	1.6e-006	0.0018
σ=5	1.4e-020	2.6e-015	5.5e-011	2.7e-007	1.7e-004	0.0128
σ=6	1.7e-014	7.2e-011	7.3e-008	2.8e-005	0.0022	0.0345

Figure 4.5: Impact of channel parameters on the probability of ping-pong effect (sampling rate of 20 Hz) [FAZK14].

By enlarging the channel parameters, the probability of ping-pong effect increases. Figure 4.5 shows that increasing any channel parameter causes higher link variability, unreliability and instability. This is the main reason for noticing higher probability of ping-pong effect when increasing either the path-loss exponent or shadow fading parameters.

Studying the channel parameters (σ,η) reveals the high dependency of the hand-off process on environmental changes. However, these values do not fluctuate significantly in indoor environments. The variation of channel parameters for both indoor and outdoor environments is shown in Table 4.1.

In an efficient hand-off algorithm, the MN should perform hand-off within a single sample (50 ms in this example). Figure 4.4 shows hand-off delay < 40 ms for all cases with $\eta < 6$ and $\sigma < 5$. This condition is always verified for indoor and outdoor environments (see Table 4.1). Thus, we get to the conclusion that **smart-HOP is potentially suitable for all environments, although for outdoor environments a user should perform a radio survey[2] to obtain a better insight**.

[2]It is difficult to predict the values of channel parameters during the experiment. A radio survey is a process that determines the radio parameter values before performing an experiment.

Table 4.1: Range of channel parameters' values in indoor and outdoor environments [ZK04].

Environment	η	σ
Outdoor	4.7 (4.30-5.10)	4.6 (2.80-6.40)
Indoor	3.0 (2.67-3.23)	3.8 (2.60-5.00)

4.2 Simulation evaluation

We have conceived a simulation model in MATLAB to further evaluate the parameters that were neither studied in the preliminary experiments nor in the probabilistic model, which are namely the link monitoring frequency (*ws*) and stability monitoring (*m*) [TAFK12, FAZK14]. We employ two APs and a MN that moves from one AP to another, similarly to the analytical set-up in Figure 4.3(a). In this model, we generate random values of RSSI at various distances from the serving AP and the neighboring AP with the aforementioned equation: $RSSI(i)[dBm] = P_t[dBm] - \overline{PL(d_0)}[dB] - 10\eta log_{10}(i/d_0) - X_\sigma[dB]$. In this simulation, user can easily decide to start and end up the Discovery Phase by upon reading the RSSI value at each distance.

Studying the impact of these parameters give us results that are very close to the probabilistic model, thus to avoid redundancy we do not present them again. However, in this model we are able to consider other (higher) values of *ws* and *m* (rather than ws=m=1). These parameters were ignored in the probabilistic model, for simplicity. In this simulation, we assume that the mobile node is initially connected to AP_a. By considering a sliding window *ws* and low threshold level T_ℓ for starting a hand-off, the MN decides for the hand-off starting slot. Then, by having the RSSI value of AP_b and considering the stability parameter *m*, the MN decides for ending the hand-off. This process repeats for $10,000$ trips and the results are averaged at the end of simulation.

Impact of *link monitoring frequency*. The *link monitoring frequency* is a parameter that is used in both the *Discovery Phase* and the *Data Transmission Phase*. In the preliminary experiments, we simply assumed *ws* = 3 for both phases. In this simulation, we study the impact of *link monitoring* in each phase separately.

1. *Impact of link monitoring frequency in the Data Transmission Phase.* By setting $ws = 3$ for the *Discovery Phase* and varying it from 1 to 5 in the *Data Transmission Phase*, we get a decreasing trend of hand-off delay for the first 3 cases and then it remains unchanged — see Figure 4.6(a). This happens to the number of hand-offs as well. This means that a small *link monitoring* value during the normal data communication of the MN, reduces the hand-off delay and the number of unnecessary hand-offs.

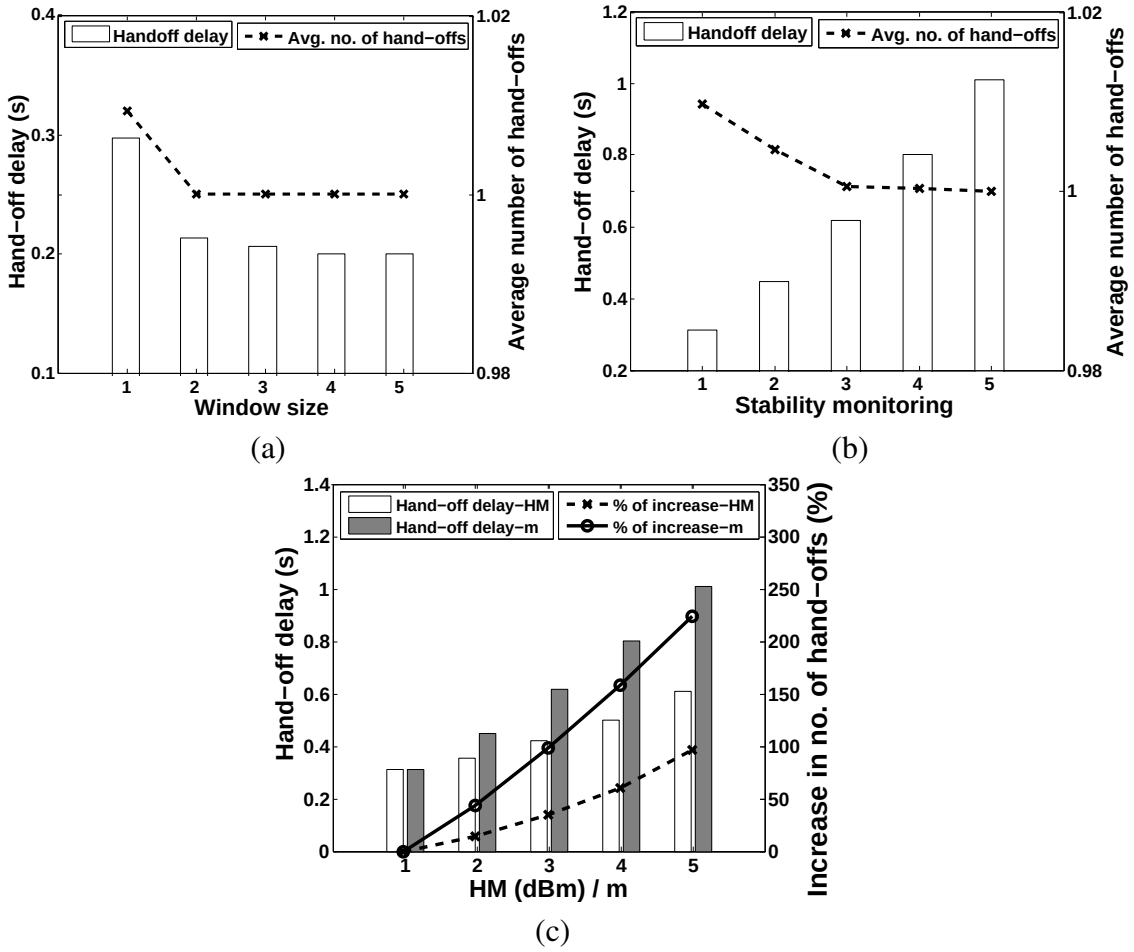


Figure 4.6: Simulation analysis: (a) Impact of window size in the *Data Transmission Phase*, (b) Impact of *stability monitoring*, (c) *stability monitoring* (in simulation model) versus *hysteresis margin* (in analytical model) [FAZK14].

2. *Impact of link monitoring frequency in the Discovery Phase.* Increasing the number of beacons for assessing the neighboring APs requires more time, which is proportional to the sampling frequency. This case is somehow similar to the stability parameter that increases the duration of link assessment. It is apparent that considering a few samples can compensate the fluctuations of random RSSI values. Hence, it is not logical to assume a large *link monitoring* value for the *Discovery Phase* due to its negative impact on the hand-off delay. In the current model with 2 APs, the result is similar to the case when changing the stability parameter, thus it is not shown here. In case of higher density scenarios the results are different, but still the trend is equal.

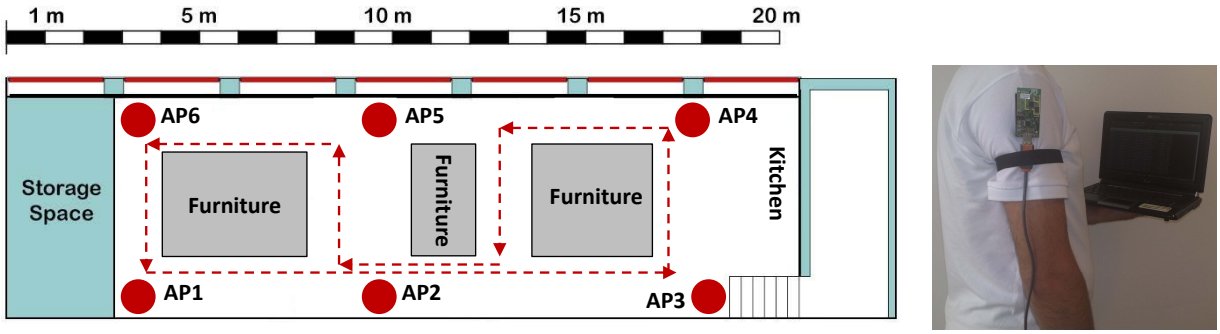
Impact of *stability monitoring*. Increasing the *stability monitoring* reduces the link variability. Each unit of *stability monitoring* adds a new *Discovery Phase*, which is composed of a set of beacons and reply packets. The results in Figure 4.6(b) indicate that the hand-off delay has an increasing trend with a high slope, which is more steep than the case of increasing *link monitoring* during the *Discovery Phase*. Considering the ping-pong effect, there is an improvement with a small stability parameter. In practice, we can substitute the stability parameter with the *link monitoring* value of the *Discovery Phase*. By this action, we can (i) reduce the link variability to eliminate the ping-pong effect and (ii) compensate the RSSI fluctuations to take accurate hand-off decisions.

Comparing correlated parameters. It is obvious that either enlarging the hysteresis margin HM or increasing the stability monitoring m reduces the link variability. It is interesting to compare the impact of these parameters before fine-tuning them for the experimental evaluation (Section 4.3). It is difficult to pick a special moment for starting a hand-off. According to the simulation model, we have a hand-off delay of 0.311 s with $m = 1$ and $ws = 3$. The period of transmitting beacons and data packets is set to 100 ms. This hand-off delay occurs at a 3.42 m distance from AP_a in the probabilistic model when $HM = 1$ dBm. By increasing the stability parameter in the simulation model and *hysteresis margin* in the probabilistic model, we compare the hand-off delays at the same point of assessment, i.e. 3.42 m. The results indicate that the hand-off delay increases more severely by adding the stability parameter rather than by adding the *hysteresis margin* — see Figure 4.6(c). This shows its severe impact on the delay that should be taken into account when designing a hand-off mechanism.

Lessons learned from the simulation model.

1. The *link monitoring frequency* in the *Data Transmission Phase* does not have a significant impact on the overall performance. Setting a small number of *ws* is the preferred, to reduce the hand-off delay.

2. Tuning the *stability monitoring* parameter enables to eliminate the ping-pong effect, but large values increase the hand-off delay remarkably. It is recommended to either choose a small value of *stability monitoring* or *link monitoring* during the *Discovery Phase*.



(a) (b)

Figure 4.7: (a) The APs' deployment in a large room, and (b) the MN attached to a person's arm and the logging PC [FAZK14].

3. The *stability monitoring* can compensate the link variability much more significantly than the *hysteresis margin*, at the cost of a higher hand-off delay.

4.3 Experimental evaluation

The *preliminary experiments* [FZA+12] revealed the best thresholds for hand-off in a controlled environment. The probabilistic and simulation analysis proved that, in theory, smart-HOP enables efficient hand-offs. At this stage, we set further experiments to enable a deeper analysis and fine-tuning of the algorithm in a more realistic environment.

4.3.1 Experimental set-up

We deployed 6 APs with a minimum power level of -25 dBm in a 80 m^2 rectangular room. The APs were attached to walls at 1.5 m height from the ground (to guarantee a better connectivity). Figure 4.7(a) illustrates the test-bed layout: position of each AP, furniture, walls and windows. Figure 4.7(b) shows the mobile node attached to the body and the logging PC[3].

As we mentioned earlier, the *low threshold level* and *hysteresis margin* parameters are very important. Instead of starting the experiment with all the 6 APs, we first confine the

[3]Initially, we connected all APs to one laptop with passive USB cables and USB2.0 hubs. Then we observed some data loss during data transfer through the UART port. Adding more PCs did not solve the problem completely. Hence, we managed to get the data log from the MN at the cost of a person carrying a laptop during the experiments.

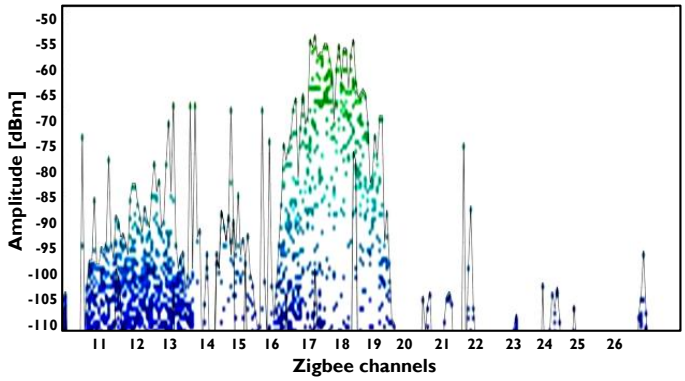


Figure 4.8: Channels activity during smart-HOP experiment in a realistic environment.

scenario to 2 APs (AP_1 and AP_2 in Figure 4.7(a)) and attached the MN to the person's arm. The sensor is attached to the shoulder, but the body eliminates Line-of-Sight (LoS) communication in some trips between AP_1 and AP_2. We refer to this set of tests as *baseline experiments*. Less APs guarantees that there is no overlapping between the neighboring APs. The person walks 4 times between these APs with a normal human walking speed (about 1 m/s).

The experimental area is a lab with at most 10 people sitting and three to five people moving sporadically and randomly in all locations. The tests were performed on Channel 26 of the CC2420 radio, which is one of the most immune to WiFi. In order to obtain a better understanding of the frequency activity, we measured the 2.4 GHz spectrum usage; a Wi-Spy spectrum analyzer [tea14] verified that there was low interference from other RF sources in Channel 26 — see Figure 4.8.

We evaluate smart-HOP in this realistic environment in two steps; (i) *baseline experiments*, using 2 APs, to further analyze the lower threshold level and the *hysteresis margin* and (ii) *extended experiments*, using 6 APs, to study the impact of *stability monitoring* and *link monitoring*. In all tests, we employ SNR-based smart-HOP as it encompasses the interference in the environment. The most relevant results are presented in the next two subsections.

Table 4.2: Description of cases considered in baseline experiments [FAZK14]

Cases	T_h	T_ℓ	HM
Case 1	-75 dBm	-76, -78, -80, -82, -84, -86, -88, -90 dBm	1-15 dBm
Case 2	-77 dBm	-78, -80, -82, -84, -86, -88, -90 dBm	1-13 dBm
Case 3	-79 dBm	-80, -82, -84, -86, -88, -90 dBm	1-11 dBm
Case 4	-81 dBm	-82, -84, -86, -88, -90 dBm	1-9 dBm
Case 5	-83 dBm	-84, -86, -88, -90 dBm	1-7 dBm
Case 6	-85 dBm	-86, -88, -90 dBm	1-5 dBm
Case 7	-87 dBm	-88, -90 dBm	1-3 dBm
Case 8	-89 dBm	-90 dBm	1 dBm

4.3.2 Evaluation - baseline experiments

Baseline experiments. In the *preliminary experiments*, we considered four groups of lower threshold levels (-95, -90, -85 and -80 dBm) and two values of *hysteresis margin* (1 and 5 dBm). The results indicated that -95 dBm is not a choice as the MN enters in the disconnected region. Now, we consider a wider range of lower threshold levels [-76, -90 dBm] increasing in 2 dBm steps, and higher threshold levels in the range of [-89,-75 dBm], which in turn generate *hysteresis margins* of 1 to 15 dBm. All the 8 cases of lower threshold levels with several HM values lead to 36 combinations. We compare all these scenarios in terms of number of hand-offs, hand-off delay and packet delivery ratio by walking four times between APs (AP_a and AP_b).

The main goal at this stage (with just 2 APs) is to pick situations that are more likely to be efficient and then reassess them with 6 APs, for further analysis. We depict the performance of smart-HOP in terms of number of hand-offs, average hand-off delay and absolute packet delivery ratio in Figures 4.9, 4.10 and 4.11, observing the following facts.

1. Selecting the lower threshold level from the lower end of transitional region with a wider *hysteresis margin* eliminates ping-pong effect — looking at Figure 4.9. Except cases (7) and (8) with narrow *hysteresis margin*, we observe acceptable situations (shadowed bars) in all other cases, corresponding to four hand-offs (the minimum for this experiment).

2. Either a very wide *hysteresis margin* or narrow margin but with a high T_ℓ causes

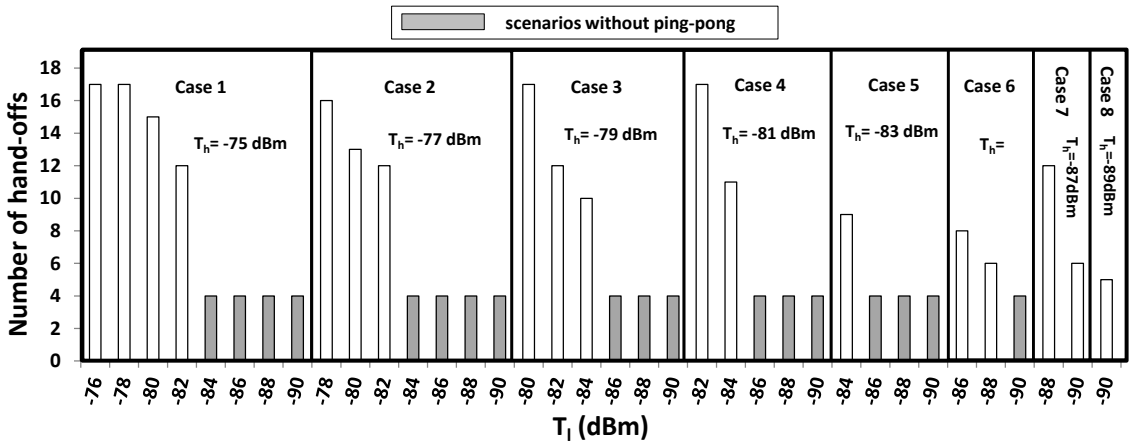


Figure 4.9: Total number of hand-offs in baseline experiments [FAZK14].

huge hand-off delay. A wide *hysteresis margin* obliges the MN to stay at the *Discovery Phase* for longer periods of time. A narrow *hysteresis margin* with high T_ℓ causes an excessive number of hand-offs that eventually enlarges the hand-off delay. In theory, an efficient hand-off (which performs fast and without ping-pong effect) should perform within one sample. For instance, if the sampling is every 100 ms, the hand-off delay should be 100 ms (for an efficient hand-off). This value is depicted by a horizontal line in Figure 4.10, as a reference. Results indicate that selecting $T_h > -83$ dBm leads to higher hand-off delays.

3. A lower hand-off delay causes higher packet delivery ratio. Observing the results of packet delivery ratio in Figure 4.11, we can make two observations; (i) increasing the *hysteresis margin* in all cases reduces the link variability and increases the packet delivery ratio, and (ii) higher values of packet delivery are achieved in situations with lower hand-off delay. The more efficient scenarios are noticed with HM between 3 and 7 dBm and T_ℓ in the range of -86 to -90 dBm.

The most efficient scenarios in the *baseline experiments* (the intersection of efficient scenarios), are further elaborated in the *extended experiments*. The *baseline experiments* reveal that with the smallest T_ℓ (-90 dBm) and $HM = 5-7$ dBm, the hand-off delay is minimum, while obtaining a maximum delivery of packets. An educated solution is

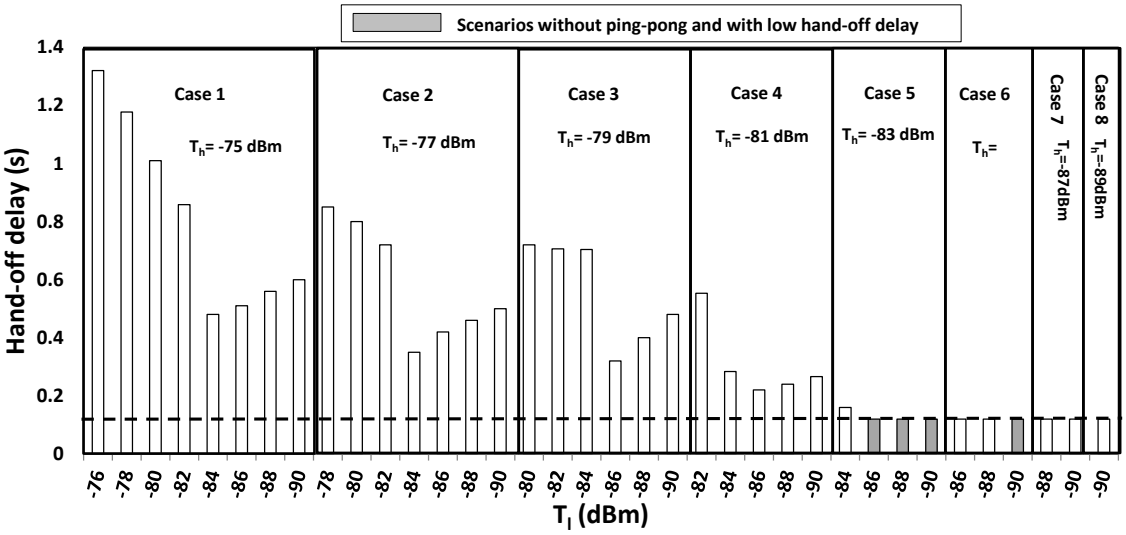


Figure 4.10: Average hand-off delay in baseline experiments. The horizontal line presents the minimum achievable hand-off delay obtained by analytical model.

to keep the MN connected to the current link as much as possible, similarly to the *preliminary experiments*. Thus, we keep the same threshold level for starting the hand-off ($T_\ell = -90$ dBm) and compare the results for several values of HM (3 to 8 dBm).

4.3.3 Evaluation - extended experiments

In the extended experiments, we increase the number of APs to six, as adding more APs creates a more realistic environment in which the mobile node experiences links overlapping (with the APs). For each set of experiments according to the selected *hysteresis margin*, the person walks in the room while the mobile node sends data periodically (every 100 ms). The person starts walking from AP_1 along the dashed line shown in Figure 4.7(a). In some parts of the way, there are obstacles that prevent Line of Sight (LoS) communication. Moreover, random movement of people creates more dynamics in the environment. For each set of experiments, the person walked for about 15 minutes (transmitting $10,000$ packets), which is about 15 full laps (dashed line circuit in Figure 4.7(a)). At the end, we computed the average hand-off delay and packet delivery ratio.

Increasing the *hysteresis margin* enlarges the hand-off delay, as it forces the MN to attach to a higher link quality AP. Figure 4.12(a) shows that the hand-off delay is minimum with 3-5 dBm *hysteresis margin* and then it features a gradual increase. The reason is

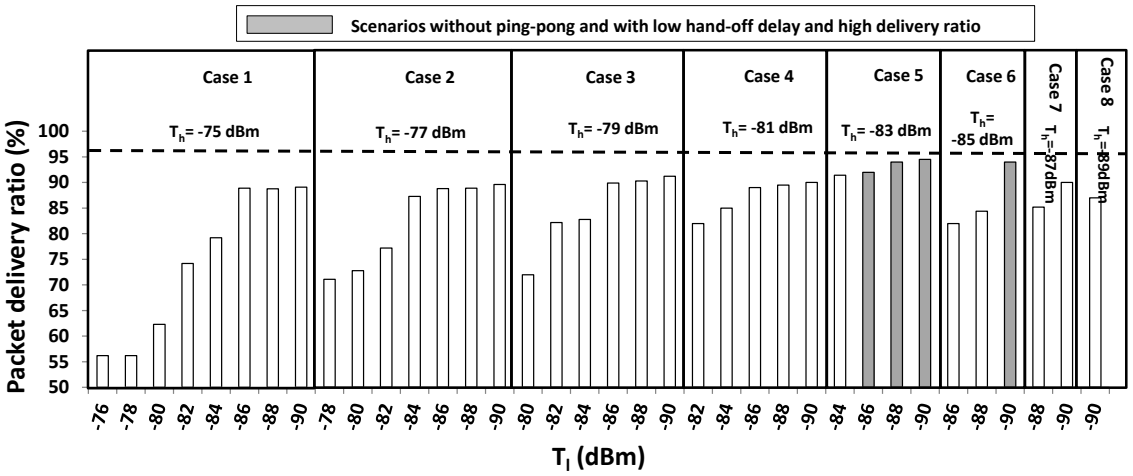


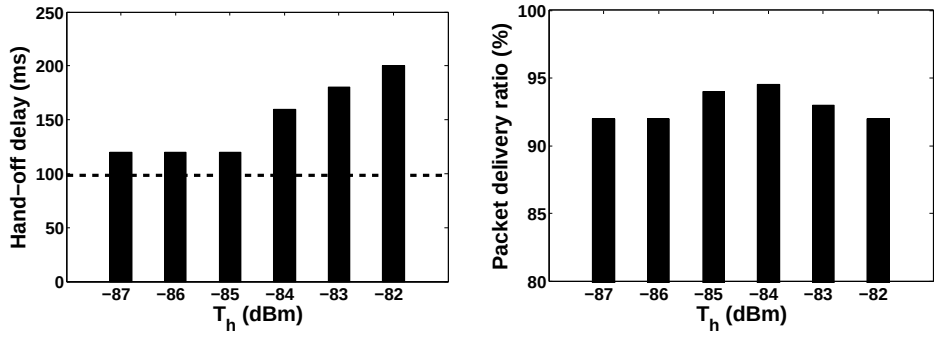
Figure 4.11: Absolute packet delivery ratio in baseline experiments. The horizontal line presents the packet delivery ratio of a simple broadcast scenario (MN broadcasts instead of performing smart-HOP).

that by enlarging the margin, the chance of staying in the *Discovery Phase* for more than one period is higher. The one period stands for the case where after sending a burst of beacons and receiving reply packets, the MN is able to observe a good link to perform the hand-off. The packet delivery ratio in Figure 4.12(b) illustrates a higher packet delivery ratio with $HM = 5$ and 6 dBm. The packet delivery rate decreases with HM since there are unnecessary hand-offs. The higher HM causes fewer packets delivered as the MN stays in the *Discovery Phase* longer.

The *hysteresis margin* should be tuned to achieve an optimal trade-off between the delivery rate and delay. By choosing the lower end of the transitional region (-90 dBm) as the lower threshold level (T_ℓ), the *hysteresis margin* of 5 dBm is the best choice.

The *stability monitoring* and the *link monitoring* parameters are the two other important parameters, which are studied in these experiments, as follows.

Increasing the *stability monitoring* increases the hand-off delay. Increasing the *stability monitoring* has a direct impact on the hand-off delay (Figure 4.13(a)). Adding one unit to m requires observing a high quality link for one more sliding window. It is interesting to notice that this raise does not have a good impact on the packet delivery since we are shrinking the *Data Transmission Phase*. Considering the fact that a slight change in stability increases the hand-off delay significantly, it is wise to tune other related parameters with less impact. Thus, we opt for choosing the minimum stability value



(a) (b)

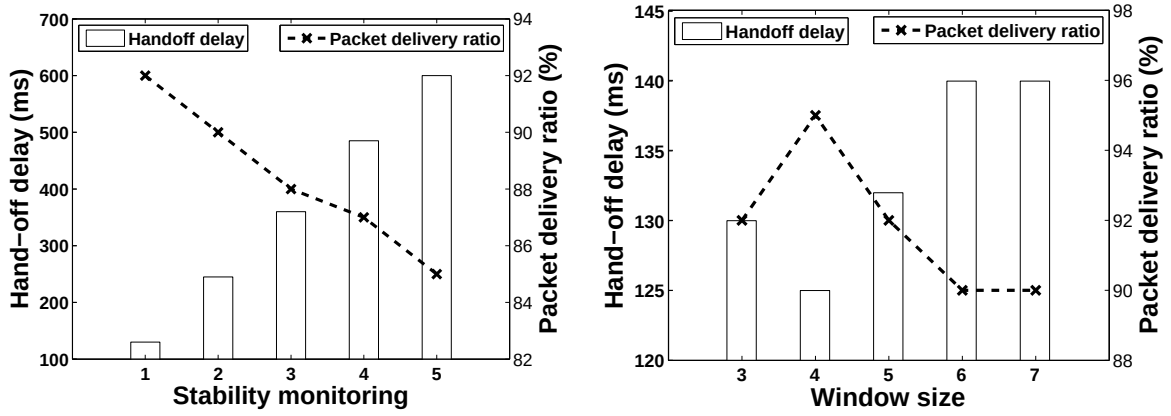
Figure 4.12: More extensive experimental evaluation with 6 APs by considering $T_\ell = -90$ dBm, (a) average hand-off delay (the horizontal line shows the optimum hand-off delay computed by the analytical model) and (b) absolute packet delivery ratio [FAZK14].

($m = 1$) for the experiment and play with the other parameters.

The *window size* parameter compensates the dynamics of the link. It defines the *link monitoring* parameter in the hand-off process, which should neither be too small nor too large. This parameter compensates the link variability and sudden RSSI changes. In a controlled environment, $ws = 3$ is selected, according to the suggestions from related work. However, in a realistic scenario, there are more sources of disturbance: (i) there is a natural variation in human gait; when a movement experiment is repeated, a person carrying a node may move a bit faster or slower than before, or may deviate slightly from the previous path; hence, a node may detect different signal strengths at the same position [FFL06]. (ii) The human body partly absorbs electromagnetic radiation, and the amount of absorbed energy depends –among other things– on the person's physique and pose, and the radio frequency [RHG$^+$11]; the results indicate that in a real environment a slightly higher window size, $ws = 4$, increases the accuracy (Figure 4.13(b)); but enlarging *ws* above this value does not improve the performance since it provides coarse grain information of the link and reduces the responsiveness of the hand-off process.

4.4 Conclusion

It is well known that the performance of a LPWN is prone to environmental conditions. Thus, we conceived a probabilistic model to investigate the impact of the most relevant



(a) (b)

Figure 4.13: (a) Impact of *stability monitoring*, (b) impact of *link monitoring* [FAZK14].

radio channel parameters on the hand-off process. We showed that the environmental changes have a direct impact on the smart-HOP performance, but it ends up performing well in various channel conditions. To have a better knowledge of the smart-HOP performance, it is recommended to perform a radio survey (to determine the path-loss exponent and shadowing standard deviation values) before the experiment. A simulation model was also conceived to verify and extend the probabilistic analysis. We studied the impact of network and channel parameters, confirming the correctness of the probabilistic analysis. The impact of *link monitoring* and the *stability monitoring* parameters were also investigated. It was revealed that the *stability monitoring* has much more strength that the *hysteresis margin* in what concerns the hand-off delay.

We performed an extensive set of experiments in a more realistic environment. These were performed in a large room with more people around, while the MN was attached to the shoulder of a person and six access points were attached to walls. A wider range of parameters was considered for the performance analysis (T_ℓ from -90 to -76 dB, HM from 1 to 15 dBm, m from 1 to 5 and ws from 3 to 7). We obtained similar parameter settings as in the initial tests, which confirmed the stability of the smart-HOP mechanism in various environmental conditions and for several network scenarios.

Overall, smart-HOP enables fast and reliable mobility support in low-power networks. It is most effective for LPWN deployments where APs have a minimum overlapping. In dense deployments, a MN is more likely to be in the connected region of one or more AP.

smart-HOP is inefficient for such dense deployments as it is tuned based on the assumption of existing transitional regions. Moreover, the single radio characteristic limits the number of MNs that can be serviced at each AP.

In the next Chapter, we will provide a literature on the mobility support in RPL and 6LoWPAN. We will also give an insight on the RPL design and functionality.

Chapter 5

An insight on RPL and on related works on mobility support

This chapter introduces the most relevant features of the 6LoWPAN/RPL standard protocol. Then, we present the most relevant mobility solutions in RPL and 6LoWPAN. We summarize the main features of these protocols and algorithms at the end of this chapter. We present our proposed mobility framework for RPL in Chapters 6 and 7.

5.1 Introduction

We have previously addressed the design and evaluation of the smart-HOP hand-off mechanism. Leverage smart-HOP for multi-hop networks, we integrate it within a standard routing protocol — RPL/6LoWPAN.

The next generation Internet, commonly referred as *Internet of Things* (IoT), paves the way for a world populated by an endless number of smart objects that are able to sense, process, react to the environment, cooperate and intercommunicate via the Internet. For over a decade, low-power wireless network research has focused on shrinking and simplifying the Internet architecture for sensor network applications. The *Internet Engineering Task Force* (IETF) has been working on protocols and adaptation layers that allow IPv6 to run over the IEEE 802.15.4 link layer. The *IPv6 over Low-power Wireless Personal Area Networks* (6LoWPAN) working group [Mul07] designed header compression and

fragmentation for IPv6 over IEEE 802.15.4 [MKHC07]. The IETF *Routing Over Low-power and Lossy networks* (ROLL) working group designed a routing protocol, referred as RPL [Win12], which is currently the *de-facto* standard routing protocol for 6LoWPAN. These standard IP-based protocols are thus a fundamental building block for the IoT.

6LoWPAN is able to support routing in the link layer and the network layer [CIC+09, HC08]. Through two routing schemes — mesh-under and route-over, respectively. In a mesh-under organization, the adaptation layer performs communication in a single broadcast domain, where all nodes can reach each other. This scheme requires link layer routing, since the multi-hop topology is abstracted by employing IPv6 support. In a route-over scheme, the routing decision is taken at the network layer, where nodes act as IP routers. Each link layer hop is an IP hop and the IP routing forwards packets between these links. The route-over routing supports a multi-hop mesh routing communication, suitable for large-scale deployments.

Two main mechanisms are employed in RPL and 6LoWPAN that partially cope with mobility. First, the periodic transmission of control packets, scheduled by the *Trickle* algorithm, can detect topological changes. During this process, RPL resumes a global routing update that causes a high overhead. Second, the *Neighbor Discovery* (ND — defined in RFC 4861 [NNS07]) mechanism, assesses neighbor reachability in a regular basis. At each activation, the ND protocol floods the entire network with router advertisements, also leading to a high overhead. A long activation interval (that reduces the overhead) leads to low responsiveness to network/topological changes. However, in the revised ND mechanism of 6LoWPAN, router advertisement packets are transmitted upon receiving router solicitation messages [SCN11].

In this chapter, first we provide a background on the RPL protocol and the processes that support mobility. Then we address the most recent mobility solutions within 6LoWPAN and RPL, which is followed by a comparison, stating the main features of the aforementioned solutions.

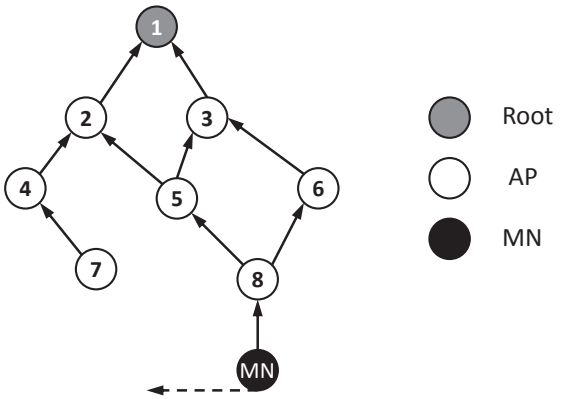


Figure 5.1: An example of having mobile node within an RPL tree, where the MN moves from the vicinity of AP_8 toward AP_7.

5.2 Basics on RPL routing

RPL is an IPv6 distance vector routing protocol that operates on top of the IEEE 802.15.4 Physical and Data Link Layers and was designed for wireless networks with very limited energy, processing and bandwidth resources. The data rate is typically low (less than 250 kbps) and the communication is prone to high error rates, resulting in low data throughput.

RPL organizes nodes in a *Destination Oriented Directed Acyclic Graph* (DODAG). Each RPL router identifies a set of stable parents, each of which is a potential next hop on a path toward the "root" of the DODAG. The preferred parent is typically selected to be the one with the lowest *rank* among the candidate parents. A network may encompass several DODAGs, which are identified by the following parameters:

– *RPLInstanceID*. This is used to identify an independent set of DODAGs that is optimized for a given scenario.

– *DODAGID*. This is an identifier of a DODAG root. The *DODAGID* is unique within the scope of an *RPLInstanceID*.

– *DODAGVersionNumber*. This parameter increments upon some specific events, such as rebuilding of a DODAG.

– *Rank*. This parameter defines the node position with respect to the root node in a DODAG.

Each node in a DODAG is assigned a *rank* that increases in the downstream direction of the DAG and decreases in the upstream direction.

RPL control messages. The RPL control messages are the new type of *Internet Control Message Protocol version 6* (ICMPv6) — defined in RFC 2463. The RPL specification defines four types of control messages:

1. *DODAG Information Object* (DIO). The transmission of this message is issued by the root node and then multicast by other nodes. This message holds the main information for constructing and maintaining a tree, e.g. current rank of a node, RPL Instance and root address,

2. *DODAG Information Solicitation* (DIS). A node that requires a DIO message from neighbors, requests it by multicasting DIS message,

3. *Destination Advertisement Object* (DAO). Each node propagates a DAO message upward (along the DODAG). Thus, this message enables the downward traffic from the root through the DODAG to this node, and

4. *Destination Advertisement Object Acknowledgment (DAO-ACK)*. This unicast message is sent by a DAO recipient to acknowledge its successful reception.

DODAG construction. In order to construct DODAGs, the *root* node initiates the process by advertising DIO messages to the neighbor nodes, as depicted in Figure 5.2. A node that receives DIO message, it computes its rank according to the objective function and then attaches to the graph. To establish downward routes, a node advertises its address by sending DAO messages toward the root node. The process of advertising DIO and DAO messages will continue until all nodes in the network join the graph. A node can also join the network by requesting DIO messages. For instantance, in Figure 5.2, Node 3 multicasts DIS messages in order to receive DIO replies. Then upon the objective function, it chooses a proper parent node and joins the graph. The construction of DODAG is mainly managed by metrics identified in the objective function. The common metric in low-power networks is a link quality estimator. The most common link quality metrics employed in RPL are MinHop [AKJ10], ETX [DCABM03] and ETOP [JEH+08]. In the

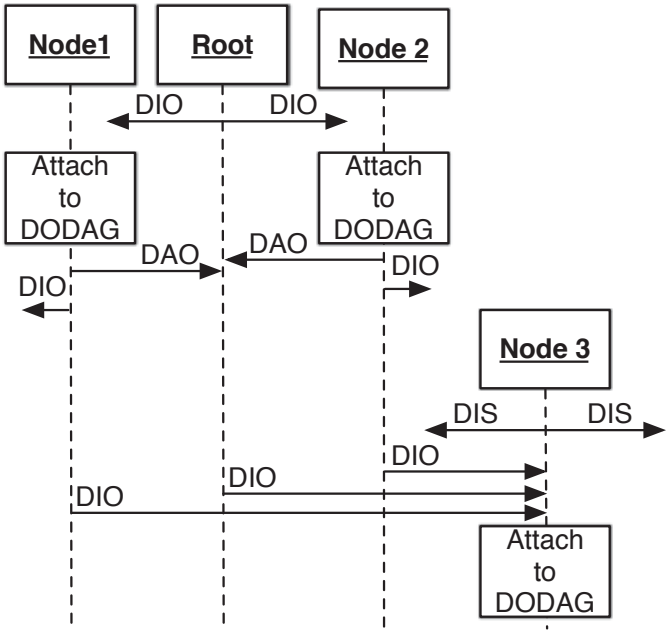


Figure 5.2: An example of DODAG construction in RPL [CMN14].

Thesis, we focus on the ETX metric, as it is the default metric in the RPL implementation available in Contiki.

Mobility detection in RPL. Mobility is indicated as one of the main sources of inconsistency in RPL [WMRD11]. Generally, there are two main approaches to tackle mobility; (i) the ICMPv6 packet transmission, controlled by the Trickle algorithm and (ii) the ICMPv6 packet transmission, controlled by the ND protocol, which are described below.

(i) RPL Trickle Algorithm. Traditional collection protocols in low-power networks typically broadcast control messages at a fixed time interval [WTC03]. A small interval requires more bandwidth and energy. A large interval uses less bandwidth and energy but topological problems may occur due to the incapability to cope with the network dynamics. The basic idea of the Trickle algorithm (defined in RFC 6206) is to propagate beacons if there is a change in routing information.

RPL reduces the cost of propagating routing states by using a Trickle-based timer [LPCS04]. Trickle is an adaptive beaconing strategy aiming at fast recovery and low overhead. While the DIS packets are sent periodically from the routers until the first parent node is selected, a Trickle timer is used to schedule the transmission of DIOs. This timer allows

the DIO intervals to exponentially increase when the network conditions are stable and quickly decrease to the minimum when noticeable changes in the network conditions are detected. The periodic Trickle timer t is bounded by the interval $[I_{min}, I_{max}]$, where I_{min} is the minimum interval defined in milliseconds by a base-2 value (e.g. $2^{12} = 4096$ ms), and $I_{max} = I_{min} \times 2^{I_{doubling}}$ is used to limit the number of times the I_{min} can double. Assuming $I_{doubling} = 4$, the maximum interval is simply calculated as $I_{max} = 4096 \times 2^4 = 65536$ ms.

The Trickle algorithm is able to maintain the topology update globally in a short period of time. A node that detects an inconsistency in a DIO message (e.g. imposed by node mobility), sets t to I_{min} and updates the tree. If the DODAG remains consistent, t is doubled each time a DIO transmission occurs until it reaches I_{max}, keeping that value constant. When the network is stable, the Trickle timer gradually converges to its maximum interval. Upon mobility, this large interval results in a very low network responsiveness. After detecting any inconsistency in the network, the DIO period of all nodes in the network exponentially decreases, affecting network overhead.

(ii) IPv6 neighbor discovery approach. RPL may use the IPv6 neighbor discovery approach [NSNS98] for detecting environmental changes. The low-power links exploit an optimized version of ND, which has been developed by the IETF as an adaptation of neighbor discovery for 6LoWPAN [SCN11]. The ND protocol allows nodes to detect neighbor unreachability and to discover new neighbors. This protocol is supported by four ICMPv6 control messages: (i) *Neighbor Solicitation* (NS): it determines the link layer address of a neighbor and verifies if a neighbor is still reachable, (ii) *Neighbor Advertisement* (NA): it replies to the NS message and it is also sent periodically to announce link changes, (iii) *Router Solicitation* (RS): it requests from the host node (mobile node in our system model) to its router, asking for information, and (iv) *Router Advertisement* (RA): a router sends periodically and as a response to the RS message that advertises its (the router's) presence with the information of the link and the Internet parameters.

In the next section, we will provide some examples on protocols that provide more efficient mobility solutions in RPL and 6LoWPAN.

5.3 Mobility support in IP-based LPWNs

Mobility support in IP-based low-power networks is a recent research topic. The 6LoWPAN has been introduced to facilitate transmission of IPv6 packets over low powered networks. It incorporates an adaptation layer between the network and data link layers. As already explained, two routing schemes are used in 6LoWPAN, depending on the layer that takes the routing decision. In the following subsections, we address some of the related works on mobility support building on the mesh-under and route-over schemes.

5.3.1 Mobility solutions within 6LoWPAN

In [MRN14, RMN12], a light version of Mobile IPv6 over 6LoWPAN is evaluated. In Mobile IPv6, movement detection is based on neighbor discovery, which is an optional feature in 6LoWPAN [NSNS98]. In this work, the authors proposed Mobinet that relies on overhearing in the neighborhood of a mobile node. By detecting a change in the neighborhood, the mobile node sends router solicitation in order to resume the neighbor discovery mechanism. Overhearing, requiring all unnecessary packets to be received by neighbor APs, which increases network overhead and consequently the energy consumption.

Listening to the mobile node neighborhood activities increases MIPv6 responsiveness for detecting the mobility event. The authors claim that the hand-off delay with the light MIPv6 over 6LoWPAN is $\approx$ 130 ms ($\approx$ 85 ms in mRPL), while the solicitation time-out for periodic router solicitation is 1 s. In mRPL, the periodicity of the mobility detection timer (that guarantees network connectivity) varies according to the network activities. In low connected networks, the timer period enlarges, which will consequently reduce the overhead.

In LowMOB [BRKY09], mobility detection is based on frequent beacon transmissions from the static nodes (APs). A mobile node joins the AP with the highest RSSI level. The mobility support is based on conventional mobile IPv6. This work also considers duty-cycled APs, where the radios are turned off intermittently. By observing a low link quality at the current AP, the corresponding AP activates the next appropriate AP.

To do so, the current AP performs a localization mechanism by using additional nodes, called mobility support points (MSPs), to find the direction of the MN.

The minimum hand-off delay of LowMOB in a single-hop scenario is $\approx$ 100 ms, which is comparable with the hand-off delay in mRPL. Using additional hardware for maintaing the mobility support is one of the disadvantages of LowMOB. In mRPL, we enable hand-off in mobile nodes without adding any hardware or changing the default RPL mechanism. The conventional MIPv6 and the localization approach that are used in LowMOB require many packet exchanges in order to detect mobility and maintain the routing. This approach requires additional hardware and high processing capabilities.

The network of proxies (NoP) [SSB12] scheme employs additional devices — proxies, which are resource-unconstrained and handle the hand-off procedure on behalf of mobile nodes (MNs). Similar to the LowMOB design, NoP requires additional hardware. Proxies are responsible for monitoring the RSSI from MNs and sharing this information with other proxies. By analyzing this information, proxies decide for the next best parent for the MN. The mobile node is programmed to send periodic ICMP packets to the proxies. Then proxies communicate with each other to maintain network connectivity by enabling and managing hand-offs. The need for periodic signaling between mobile nodes and proxies and also among proxies increases network overhead and energy consumption. In mRPL, the beaconing stops if there is some activity in the medium. Moreover, neighbor APs reply only if there is a need for hand-off. The minimum hand-off delay in NoP is $\approx$ 117 ms, which is higher than the mRPL hand-off delay.

5.3.2 Mobility solutions within RPL

In [LSD+12, LSN+12], the authors focus on mobility support in RPL. The system model assumes a fixed set of nodes, while MNs get access to the fixed nodes directly or via multiple hops, through other MNs. The mobility detection is obtained by employing a fixed timer (instead of the Trickle timer). The authors conclude that with higher DIO transmission, the connectivity increases, at the expense of additional overhead. To increase network responsiveness, DIO packets are transmitted more frequently. This method increases network overhead and energy consumption. In mRPL, the periodicity of DIO

transmission is tuned based on the data transmission period, to minimize the overhead.

Upon finding a new neighbor, immediate probing updates the ETX value to select the preferred parent in a timely fashion. To avoid loops after hand-off, the child nodes are discarded from the parent set. The proposed model has high overhead in terms of fixed and periodic beaconing. Moreover, it disables the Trickle timer, which is very useful in the absence of mobility. The evaluations are performed on a vehicular ad-hoc network (VANET) with a 24 Mbps transmission rate and a minimum of $\approx$ 11 m/s speed. The results indicate a packet delivery rate of about 80% for a scenario with one MN and one AP (compared with $\approx$ 100% in mRPL), which increased to $\approx$ 100% for a scenario with 10 MNs. This is because increasing the number of mobile nodes in a region increases network connectivity.

ME-RPL [KBBAS12] assumes that mobile nodes are identified by static RPL nodes. By enabling a learning algorithm, nodes that change their parent more often query their neighbors with lower DIS intervals. It means that the DIS interval is dynamically adapted according to the network inconsistencies. This strategy is also used in mRPL to dynamically adjust the beaconing interval and reduce network overhead. The MNs select static nodes with high quality links as their best parents.

In ME-RPL, sudden movements are not detected in real-time, since a learning algorithm is used. Thus, the problem of low responsiveness of the RPL routing to cope with environmental changes and inconsistencies still exists. In mRPL, a timer assures the timely detection of sudden movements (i.e. mobility detection timer). Tuning this timer guarantees network responsiveness. In ME-RPL, the maximum packet delivery ratio is $\approx$ 85%, which is outperformed by mRPL (roughly $\approx$ 100%).

MoMoRo [KC14] supports mobility in a sparse traffic network that runs RPL. Mobility is managed by an additional layer between the data link and network layers. After a packet transmission failure, MoMoRo makes one more attempt to reach the destination by transmitting a unicast packet. If it fails again, MoMoRo starts searching for a new route by broadcasting beacons and collecting replies from neighbors. The results of MoMoRo evaluations show that it improves the packet delivery ratio up to 85% comparing to the default RPL. However, it increases network overhead drastically, from 6 pkt/min in RPL

to 65 pkt/min in MoMoRo. In mRPL, we achieved $\approx 100\%$ packet delivery ratio with only 22% additional overhead in terms of number of control messages compared with RPL routing[1].

In MoMoRo, mobility detection depends only on the packet loss. This passive approach makes the network very low responsive to topological changes caused by mobility. Moreover, in low-power networks, it is very usual for the links quality to drop temporarily, causing packet loss, so a hand-off decision based on packet losses may lead to unnecessary route maintenance that consequently increases network overhead.

5.4 Conclusion

In this chapter, we explained the main features of the RPL routing protocol together with the existing mobility solutions in RPL. Table 5.1 summarizes the performance of related works and compares them with our proposed mechanisms, mRPL and mRPL+, which will be elaborated in Chapters 6 and 7. We conclude that in terms of packet overhead and responsiveness, mRPL and mRPL+ outperform other mobility solutions.

[1]The 22% additional overhead is an average value calculated with different RPL settings compared with the lightest RPL setting (with Trickle parameters of $< 12, 8 >$). More details are provided in Chapter 6.

Table 5.1: Comparing mobility solutions in IP-based low-power networks. In this table MU and RO represent mesh-under and route-over respectively. The main metrics are additional Hardware, Overhead (OH) and Responsiveness (Resp.). "+" indicates low packet overhead or high responsiveness. Regarding the additional hardware (HW) requirement, we state "Y" if there is a need for extra HW and "N" if there is no need.

Reference	Routing Mechanism	Mobility Detection	Mobility Solution	Extra HW	Performance	
					OH	Resp.
Light MIPv6 [MRN14, RMN12]	MU	overhearing	Light MIPv6	N	-	+
LowMOB [BRKY09]	MU	Periodic Beaconing	MIPv6	Y	-	+
NoP [SSB12]	MU	Periodic Beaconing	MIPv6	Y	-	+
RPL for VANETs [LSD$^+$12, LSN$^+$12]	RO	Fixed DIO	Immediate ETX update	N	-	+
ME-RPL [KBBAS12]	RO	Trickle	Adaptive DIS	N	+	-
MoMoRo [KC14]	RO	Packet Loss	Immediate Beaconing	N	-	-
mRPL [FMA14b]	RO	Timers + Trickle + Data Pkts	Immediate Beaconing	N	+	+
mRPL+ [FMA15]	RO	Timers + Trickle + Data Pkts	Overhearing	N	+	++

Chapter 6

mRPL design and evaluation

In this chapter, we explain the smart-HOP integration within RPL (dubbed mRPL). We describe the low responsiveness and high overhead of the two native algorithms in RPL that handle network topology changes (Trickle and Neighbor Discovery). We will show by simulation and experiments that mRPL halves the packet loss and reduces the hand-off delay dramatically to one tenth of a second with a sub-percent additional packet overhead. The contribution in this chapter is mainly supported by publication [FMA14b].

6.1 Introduction

There are four main RPL features that motivated us to grant it with mobility support:

1. the proactive feature of RPL that generates and maintains stable routing tables; a periodic broadcast of control messages among all nodes maintains the paths and link states between them; in reactive routing protocols; such as AODV [PR99] and DSR [JMB+01], routes are established upon request, so they do not respond quickly to environmental changes due to mobility or link degradation; RPL maintains the route in the background with minimal overhead; moreover, for an application with limited mobility requiring an infrastructure, RPL is very suitable

2. unlike other proactive routing protocols (e.g. OSPF [PSH+02]), RPL exchanges local information among neighbors to repair routing inconsistencies, instead of globally advertising control messages,

3. RPL runs a tree-based structure that is suitable for data collection WSN applications and

4. the IPv6-based addressing in RPL naturally performs the interoperability with other Internet devices.

Contributions. The main contributions proposed in this Chapter are:

1. Efficient hand-off mechanism for RPL with good performance, correctly delivering nearly 100% packets with at most 90 ms hand-off delay and $< 1\%$ additional overhead upon nodes' mobility in high traffic scenarios;

2. Smooth integration and backward compatibility with the standard RPL/6LoWPAN; The proposed mobility solution keeps the standard RPL protocol unchanged while providing backward compatibility with the standard implementation, i.e. standard and smart-HOP-enabled nodes can coexist and inter-operate in the same LPWN;

3. Collision avoidance mechanism (to avoid collision during the hand-off process while collecting packets from neighbor APs) and loop avoidance mechanism (to avoid closed loops in RPL routing upon mobility);

4. Implementation over a SOTA operating system (Contiki), for which the open source is freely available [Pro14a].

5. Simulation (Cooja) analysis and experimental validation with commodity hardware platforms in a realistic environment;

6.2 mRPL overview

As previously mentioned, we are integrating smart-HOP within RPL in a way that is very simple, effective and backward compatible with the standard protocol. In this model, the standard RPL protocol is unchanged while providing mobility support, i.e. standard and smart-HOP enabled nodes can coexist and inter-operate in the same network.

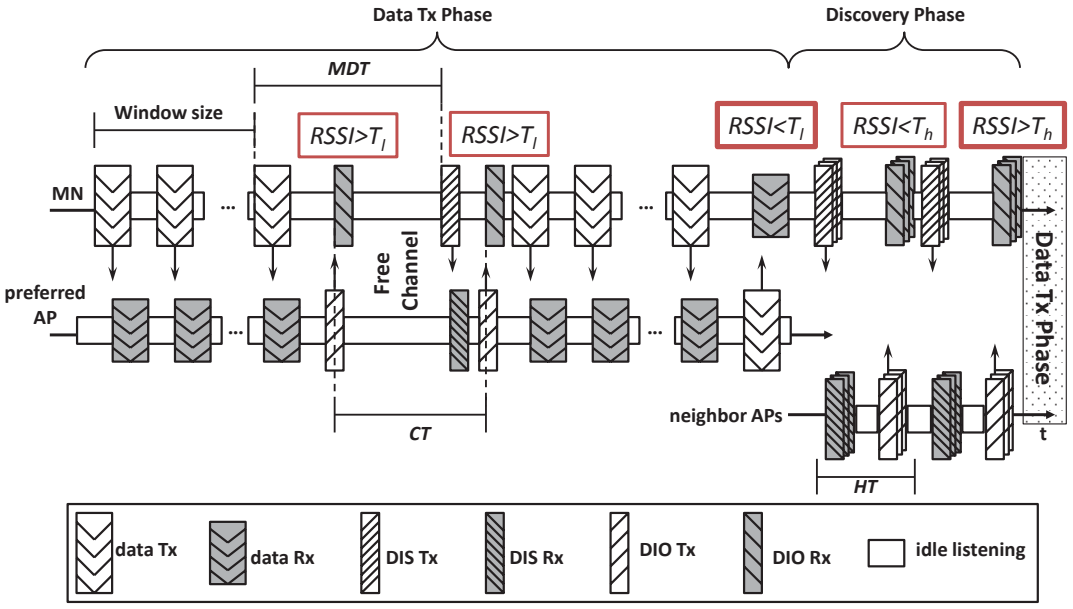


Figure 6.1: mRPL timing diagram for the *Data Tx Phase* and *Discovery Phase*.

The general procedure of beacon and data exchanges in smart-HOP integrated in RPL (mRPL) is similar to the original smart-HOP design, except employing RPL control messages (DIS and DIO) as beacons and adding some timers to improve reliability and efficiency. The timeline of the algorithm is depicted in Figure 6.1. In this approach, the MN gets a reply packet (unicast DIO message) immediately after transmitting a predefined number of data packets (*window size*). The DIO reply message (that holds the average RSSI level), implicitly filters out the asymmetric links.

Upon detecting a good quality link (from the average RSSI level in the DIO reply message), the MN continues the *Data Transmission Phase*. By observing the ARSSI degradation, MN starts the *Discovery Phase*. MN resumes the data communication with the serving AP until finding a better AP. After a successful hand-off, the nullifying process of the RPL algorithm is executed[1].

To assess the potential parents, the MN broadcasts a burst of DIS control messages. Then, all neighbor APs reply to the MN in a non-conflicting basis (this will be discussed in detail in Section 6.3). The average RSSI level is embedded in the unicast DIO reply. Upon reception of each DIO reply, MN compares the ARSSI value with the T_h level. If it is not satisfactory (ARSSI below T_h), MN continues broadcasting DIS bursts periodically

[1]Parent nullifying is a process in which the preferred parent is removed and the *rank* is set to infinity.

(with respect to the *Hand-off Timer*). Upon detecting a high link quality (ARSSI above T_h), the *Discovery Phase* stops and the MN resumes regular data communication (with a new preferred parent) — *Data Transmission Phase*.

6.3 mRPL in detail

This section details the mRPL design. We first describe the additional timers that improve the efficiency and reliability of the hand-off process. The enhanced control message packets and the priority assignment of reply packets (DIO messages from neighbor APs) are then described. The Trickle timer setting (during the hand-off) and the parent selection are also discussed.

Timers. We have implemented four main timers to facilitate an easier and faster perception of the link degradation and the parent unreachability. The use of these timers within the *Data Transmission* and *Discovery Phase* is instantiated in Algorithms 1 and 2 and briefly described as follows.

(i) *Connectivity timer (T_C).* Mobile nodes constantly monitor the channel activity to detect any packet reception from their serving AP. Every MN runs a timer to increase the RPL routing responsiveness — the connectivity timer. The periodicity of the connectivity timer is set according to the maximum Trickle interval (I_{max}). During this period, the MN keeps listening to the channel and monitors the incoming packets from the serving parent. Upon elapsing T_C, if the MN observes a silent parent, then it starts the *Discovery Phase*. Upon detecting any packet reception from the serving AP (e.g. Trickle DIO, unicast DIO or a data packet), connectivity timer is reset.

(ii) *Mobility detection timer (T_{MD}).* Periodic DIS beaconing of the MN requests a unicast DIO message from the serving AP. The MN reads the ARSSI level related to the DIO message to assess the reliability of the link. Moving a node or appearing an obstacle between two nodes may result in losing the request or reply packets. In this situation, the MN starts the *Discovery Phase* to find a new serving parent. The periodicity of the mobility detection timer is set according to the data generation rate at the MN.

(iii) *Hand-off timer (T_{HO}).* It is paramount to reduce the hand-off delay. This timer manages the periodicity of broadcasting bursts of DIS to the neighboring parents. This period should enable to accommodate transmitting bursts of DIS with the highest possible rate and receiving intermittent replies from neighbor nodes. The DIO replies are collected immediately after sending each burst. The sequence of sending replies by each parent is scheduled in such a way to reduce the probability of collision.

(iv) *Reply timer (T_R).* The serving parent is supposed to reply to the MN by unicasting a DIO control message at certain instants. Selecting a wrong moment to reply may cause a collision with the data packets, which in turn triggers the *Discovery Phase*. The parent node extracts relevant information from the packets that are received from the MN (e.g. data packet counter in each window size). The reply time is calculated by $(ws - C) \times T_{DIS}$, where C represents the counter of DIS packets within each window size (ws) and T_{DIS} indicates the DIS interval. This reply time is adaptively changing upon receiving new packets.

Algorithm 1: smart-HOP in mRPL: *Data Transmission Phase*

```
1  begin
2      if received DIO packet then
3          reset T_C;
4          if ARSSI < T_ℓ then
5              go to the Discovery Phase;
6          else
7              continue the Data TX Phase;
8          end
9      else if T_MD expires then
10         reset T_MD;
11         unicast burst of DIS;
12         go to the begin;
13     else if T_C expires then
14         go to the Discovery Phase;
15     end
16 end
```

Enhanced control messages. To integrate the smart-HOP algorithm within RPL, we enhanced the RPL control messages rather than creating new ones. This approach guarantees backward compatibility with the standard RPL, i.e. standard RPL nodes can coexist

Algorithm 2: smart-HOP in mRPL: *Discovery Phase*

```
1  begin
2      if received unicasted DIS message then
3          store RSSI readings;
4          store counter value C of the latest DIS packet;
5          reset T_R with (ws − C) × T_DIS;
6          if T_R expires then
7              calculate average RSSI;
8              send unicast DIO message with average RSSI;
9          else
10             continue Discovery Phase;
11         end
12     else
13         continue the Data TX Phase;
14     end
15 end
```

and inter-operate with smart-HOP-enabled nodes in the same network. We provide details on the packet formats of control messages in Appendix B.

RPL control messages are transmitted on a regular basis; however, during the hand-off process, they follow specific rules. In the *Data Transmission Phase*, the DIS is sent from the MN to the AP (unicast) and the preferred parent replies with a unicast DIO. The type of DIS and DIO is detected by reading a flag that reflects the status of each node (will be explained next). In the *Discovery Phase*, the MN multicasts DIS messages to all neighboring APs and receives unicast DIO replies.

smart-HOP enables transmitting unicast DIS control messages to probe the serving AP in order to ensure the parent is reachable and reliable (RPL transmits multicast DIS and DIO packets). To distinguish between the mRPL DIS and the native RPL DIS, a one bit flag (*F-DIS*) is implemented. Initializing this field to "0" represents the multicast transmission of the RPL DIS. Instead, setting this field to "1" reflects the unicast mRPL DIS transmission. The additional two bits of "*C*" describe the counter of DIS messages within a window size. In mRPL with $ws = 3$, the counter increments to a maximum of 3.

The mRPL DIO message adds two fields: (1) *F-DIO* that stands for the flags and (2) *ARSSI* that holds the average RSSI reading at the potential parent node. The two bits of *F-DIO* distinguish three cases: (i) *F-DIO*=0 corresponds to the RPL DIO, (ii) *F-DIO*=1

Table 6.1: The priority assignment to APs according to the ARSSI values.

Priority	Range of average RSSI readings
$prio = 1$	$-85 < ARSSI < -80$ dBm
$prio = 0$	$ARSSI \geq -80$ dBm

indicates the mRPL DIO within the *Data Transmission Phase*, and (iii) *F-DIO*=2 reflects the mRPL DIO within the *Discovery Phase*.

Priority assignment. In order to reduce the packet collision during the *Discovery Phase*, we prohibit some of the APs to reply to the MN; parents with $ARSSI < T_h$ are excluded from the candidate parents' set and do not reply. To do this, each parent assigns a priority according to the average RSSI readings, as shown in Table 6.1. The priority assignment schedules the DIS transmissions in different slots. Since LPWNs are likely to operate in the transitional region, it is more likely that different parents choose the same slot. A timer schedules the DIS transmission (t_{offset}) after detecting a busy channel as follows as expressed in Equation 6.1.

$$t_{offset} = (ws - C) \times T_{DIS} + t_2 \times prio + rand(t_1, t_2) \tag{6.1}$$

The first part of this equation, $(ws - C) \times T_{DIS}$, is the waiting time for receiving the complete DIS messages transmitted, which is similar to the *Data Transmission Phase* of the smart-HOP algorithm. We force the higher quality APs ($prio = 0$) to transmit earlier ($t_2 \times prio = 0$ ms) and the lower quality APs ($prio = 1$) transmit later ($t_2 \times prio = t_2$ ms). A random delay $rand(t_1, t_2)$ is also added to reduce the possibility of collisions deriving from APs with the same priority level. It is important to note that with $t_2 \times prio$, lower quality links wait at most for t_2 ms, ($max((prio = 0) \times t_2 + rand(t_1, t_2)) = t_2$), which is measured by $(prio = 1) \times t_2 = t_2$ ms. The random delay also reduces the possibility of collision between the replies from the lower quality and the higher quality APs. Random values are set to 10 and 15 ms, which are above the maximum possible transmission rate[2]. Considering $ws = 3$ and $T_{DIS} = 15$ ms, in the worst case (i.e. $rand(t_1, t_2) = 15$ ms),

[2]We use Tmote Sky motes that are equipped with the Chipcon 2420 radio chip [Ins07], operating at 2.4 GHz with 250 kbit/s data rate. The packet size depends on the data payload, which is added to the header

it takes at most 75 ms for the MN to get all replies from the neighboring APs. In our system model, we are considering a wise deployment of APs in order to avoid very high or very low density of APs. Our tests provide minimal overlap between contiguous APs, preventing the possibility of having multiple high quality APs in range.

Trickle setting during hand-off. According to the Trickle algorithm, all nodes (roots/routers) broadcast messages (DIOs) to exchange information with the neighbor nodes. The transmission interval is bounded and enlarged upon network stability. When a node moves, it interferes with the network stability and hence the interval is set to its minimum value (I_{min}). We keep the Trickle interval unchanged during the hand-off process, while keeping the transmissions' schedules independent. As already mentioned, the *F* fields of the control messages (*F-DIS* and *F-DIO*) are added to distinguish between mRPL and RPL messages.

Loop avoidance mechanism. In RPL, when a node disconnects from its parent, the *rank* value is set to infinity. This enables the MN to connect to any neighboring node, even the ones with a lower *rank*. For instance, the MN may select a neighbor node that was previously the MN's child (before the hand-off) as the new parent. Since the neighbor has a lower *rank* compared with the infinity, according to the default RPL, the MN is allowed to choose it. As shown in Figure 6.2, in DODAG 1, Node 5 (MN) has a parent (Node 2) and three children (Nodes 7, 8 and 9). Each node delivers data to a lower rank node (written besides each node — R1, R2, ...). When Node 5 moves out from the range of Node 2, according to the RPL routing, DODAG 2 is established. In this case, first the MN's rank is set to infinity and then it picks a neighbor with the highest ARSSI level and the lower rank level (6). Thus, Node 7 (Node 5's previous child) is selected as the preferred parent. The data messages from Node 5 are forwarded to Node 7, and Node 7 forwards to Node 5 (its parent), which represents a closed loop.

RPL has some loop detection mechanisms; however, loops cannot be fully avoided and thus may still occur. To fix this, RPL performs global repairs, reconstructuring the

and footer. Since RPL runs an IPv6 addressing strategy, we assume that the packet size is 127 bytes in the worst case. Considering the radio data rate and the packet size, the node is able to transmit at most 246 packets/s (1 packet every 4 ms). The propagation delay, modulation, demodulation, fragmentation and de-fragmentation extend this approximate transmission delay. In real world experiments, it is wise to pick intervals larger than 4 ms to ensure successful transmissions.

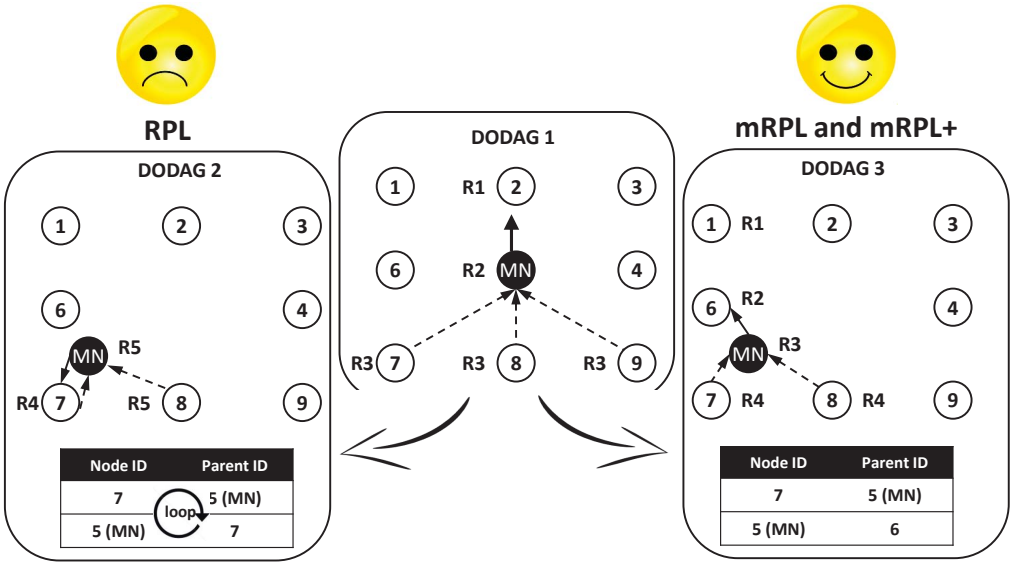


Figure 6.2: The DODAG 1 updates upon mobility. The DODAG 2 updates by applying the standard RPL algorithm, increasing in a closed loop. The DODAG 3 updates according to mRPL, avoiding the closed loop. Rank is specified besides each node (R1, R2, ...)

routing tree, updating the rank of all nodes in a DODAG. This behavior is not efficient as a MN will need to start the whole process of finding a new parent again, which is highly time and energy consuming.

In this context, we devised a simple yet efficient loop avoidance mechanism. We analyzed two different approaches to avoid the loop effect. In the first, the MN gets replies from all neighboring APs and then ignores the messages from the previous children. Thus, after creating the set of alternative parents, the children are excluded from the set. In the second, the children decline to reply the previous parent's request for joining. We select the latter approach as it leads to less communication and processing overhead during hand-off. DODAG 3 in Figure 6.2 shows a scenario where Node 5 disconnects from Node 2 and connects to Node 6, avoiding to choose one of its previous children.

Memory overhead. The memory overhead of the standard RPL against mRPL is illustrated in Table 6.2. smart-HOP has been integrated with about 4 kB ROM and 1 kB RAM extra, representing 10% of additional memory footprint.

Table 6.2: Memory usage in standard RPL versus mRPL

Implementation	ROM (bytes)	RAM (bytes)
RPL (MN)	40,202	7,660
mRPL (MN)	44,348	8,562
RPL (AP)	40,336	7,606
mRPL (AP)	44,022	8,512

6.4 Simulation analysis

We implemented and tested the protocol with the Cooja simulator, which easily ports to the sensor hardware and provides the opportunity of analyzing different network conditions. Since low-power wireless links are very prone to external radio interference from other wireless technologies operating in the ISM band, simulators are usually unable to provide a very accurate radio interference model. Each indoor/outdoor environment exhibits specific link behaviors that are impossible to mimic in the simulated environment. mRPL has been designed to perform well in networks with full AP coverage and minimum overlap between neighboring APs. In simulation, we are able to establish an environment that provides these requirements, but in real experiments, links may overlap differently and more dynamically (more or less). We will compare simulation and experimental results in Section 6.5 to show the necessity of performing experimental tests in order to enrich the radio propagation and interference models in simulation.

6.4.1 Simulation setup

In order to implement and evaluate mRPL, we opted for the Contiki 2.6.1 [DGV04] operating system (OS), which supports the Cooja simulator. The main reasons for selecting Contiki are: (i) the availability of a RPL/6LoWPAN implementation that is reasonably mature and widely used, (ii) the ease of porting Cooja code to commodity hardware platforms, and (iii) the availability of a mobility plugin in Cooja [Ost14], that enables to evaluate mRPL in a repeatable environment[3].

[3]By default, Cooja does not support mobility. Nevertheless, based on the fact that each deployed mote has its own location represented in a two-axis (x,y) system, a Cooja mobility plugin [Ost14] was developed that is capable of loading specific mobility trace-files using the Interval Format.

Table 6.3: Description of the RPL scenarios used in simulation.

Scenarios	I_{min}	$I_{doubling}$	DIO_{min}	DIO_{max}
(12-8)	12	8	4.096 s	1048.576 s
(12-1)	12	1	4.096 s	8.192 s
(10-2)	10	2	1.024 s	4.096 s
(8-1)	8	1	0.256 s	0.512 s

In this section, we compare mRPL with different settings of the standard RPL, considering different topologies. Then, we study the impact of other parameters on mRPL performance. The major parameters that impact RPL performance are I_{min} and $I_{doubling}$ in the Trickle algorithm. We considered four RPL scenarios by varying the tuple $< I_{min}, I_{doubling} >$ values, as defined in Table 6.3. The evaluation focuses on the impact that these Trickle parameters have on the following network metrics:

Hand-off delay. It represents the average time required to perform the hand-off process in mRPL or the time spent to discover a new preferred parent in the standard RPL.

Total packet overhead. We identify all the non-data packets (control messages) as network overhead. RPL uses ICMPv6 based control messages (DIS, DIO and DAO) for building and maintaining DODAGs. mRPL utilizes these control messages to detect the mobility and perform the hand-off process.

Packet delivery ratio (PDR). It is defined as the number of successfully received packets at all APs over the total number of packets sent from MNs. The successful delivery rate of mRPL is compared with different RPL scenarios in the presence of mobility.

Total DAO packets. To establish downward routes, RPL nodes send unicast DAO messages upward. The next hop destination of a DAO message is the preferred parent. After switching to the best parent, the child node informs the previous parent about its disconnection and the selected parent about its reachability. The total number of DAO packets is an indication for assessing the routing responsiveness and the number of hand-offs in a mobility-enabled network.

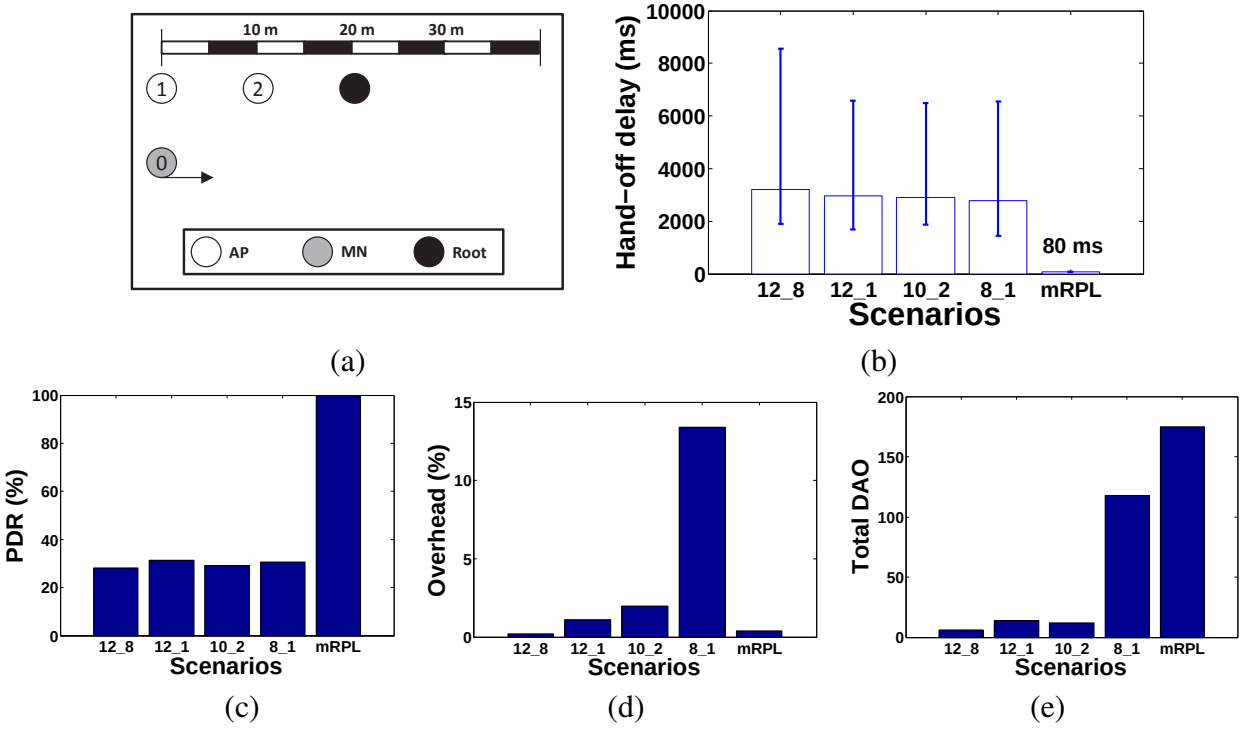


Figure 6.3: Simulation results for a network topology with two APs. (a) simulation scenario, (b) hand-off delay, (c) packet delivery ratio, (d) total overhead in terms of control messages, and (e) total number of DAOs.

6.4.2 RPL vs. mRPL

To evaluate the proposed algorithm, we consider three network topologies: (1) with two APs, (2) with four APs deployed in a row, and (3) with eight APs deployed in two parallel rows. In the first deployment with two APs (Node 1 and Node 2, 10 m apart — see Figure 6.3(a)), the MN travels 15 times between AP_1 and AP_2 with a constant speed ($v = 2$ m/s) and transmission power of -25 dBm, while generating data at a 30 pkt/s rate. Similarly, in the two other deployments (Figures 6.4(a) and 6.5(a)), the MN moves from one left side to the right side with the same constant speed and then returns back to the starting point.

Connectivity is guaranteed by providing a fast and reliable hand-off process. mRPL is able to detect and perform a hand-off within tens of milliseconds (80 to 83 ms), which is much faster than all RPL scenarios — see Figures 6.3(b), 6.4(b) and 6.5(b). We have estimated the hand-off delay in the standard RPL as it does not have a hand-off mechanism. *The "hand-off" in RPL is assumed to start at the moment when packets start*

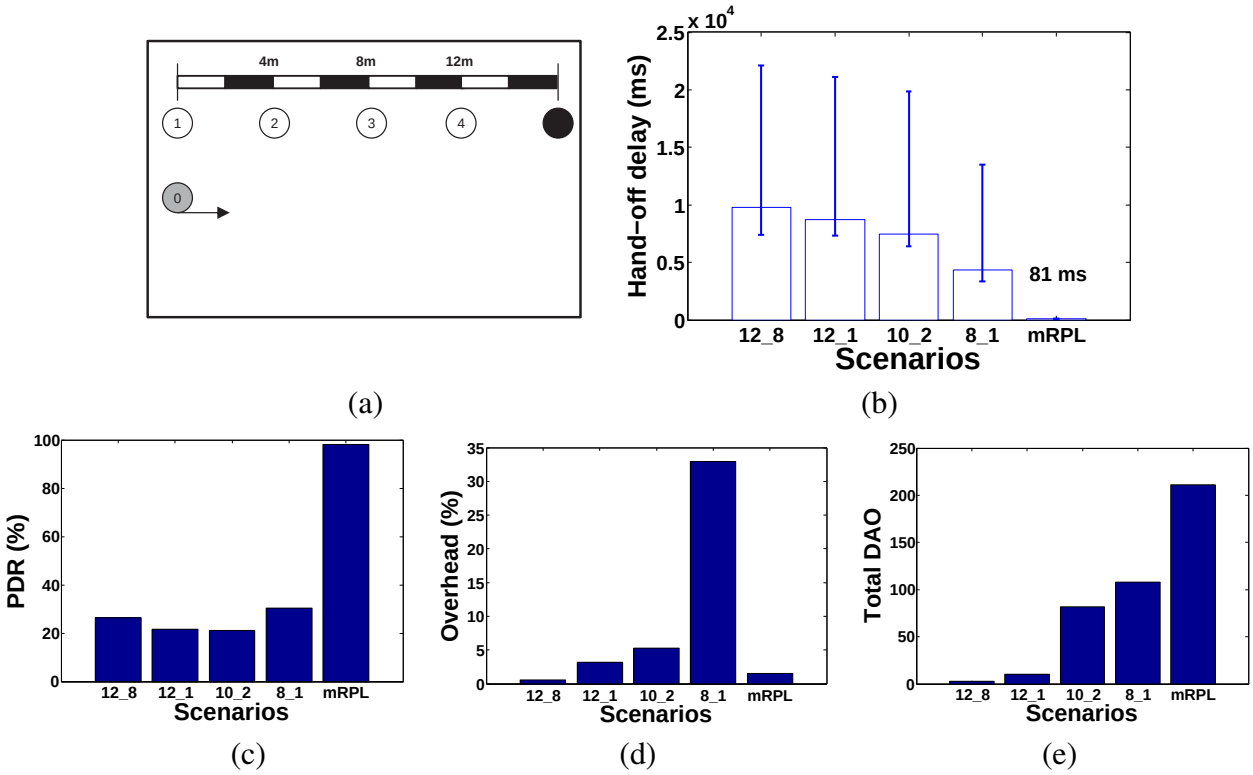


Figure 6.4: Simulation results for a network topology with four APs. (a) simulation scenario, (b) hand-off delay, (c) packet delivery ratio, (d) total overhead in terms of control messages, and (e) total number of DAOs.

to get lost at the serving parent and to finish when the new parent starts to successfully receive data packets from the MN. The high data generation rate accelerates the updating of the ETX metric that leads to a fast parent switching process during link degradation.

The hand-off delay of RPL scenarios fluctuates a lot, as the mobility detection mechanism depends on various conditions (e.g. data rate, Trickle timer and ND protocol); thus, the responsiveness to environmental dynamics is not guaranteed in RPL. The average hand-off delay in RPL scenarios varies from 2776 ms to 9776 ms in these three network topologies (Figures 6.3(b), 6.4(b) and 6.5(b)). In RPL, the mobile node switches between parent nodes in its parent set. In order to update the parent set information, it uses the Trickle and ND algorithms. The Trickle algorithm (that schedules the control message exchanges) will enlarge intervals if the network is stable. To detect mobility in this condition, the RPL node either waits for receiving the NA message or requests this message by multicasting the NS message to its neighboring APs. These messages are supported by the ND protocol to detect parent unreachability. The major drawbacks of

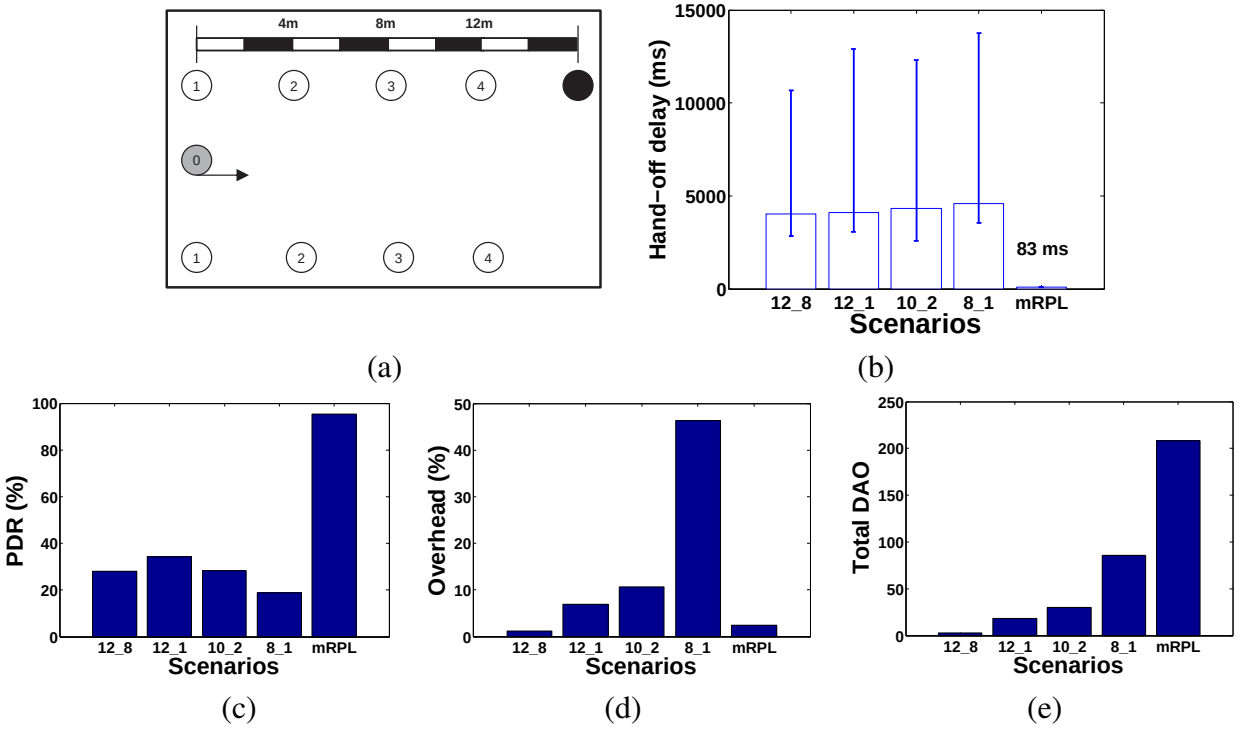


Figure 6.5: Simulation results for a network topology with eight APs. (a) simulation scenario, (b) hand-off delay, (c) packet delivery ratio, (d) total overhead in terms of control messages, and (e) total number of DAOs.

RPL concerning network connectivity are: (i) the sudden changes due to nodes mobility are not quickly detected if the network has been stable for a while, (ii) the ND protocol is initiated at the parent side (like a passive hand-off), which enlarges the hand-off duration, and (iii) resuming the ND protocol (that reconstructs the routing trees) is very expensive.

mRPL is able to provide nearly 100**% packet delivery ratio.** A fast hand-off process enables transmitting most of the packets to the target access point. In RPL, the MN should wait for control messages from the nearest AP. A longer delay causes more packet losses as the MN is not connected to any AP. In mRPL, the MN is able to send data to the previous parent during the *Discovery Phase* until finding a new preferred parent. This mechanism increases the chance of delivering most of the data packets, as shown in Figures 6.3(c), 6.4(c) and 6.5(c).

The control message overhead of mRPL is comparable with the RPL settings with minimum overhead. In RPL, after creating DODAGs during an initialization phase, if the network remains stable, the periodicity of control message exchanges will converge

to its maximum value. For instance, according to Table 6.3, in the $< 12, 8 >$ RPL scenario, the periodicity is 1048.576 s and with $< 8, 1 >$ is 0.512 s. Again, note that a higher message transmission rate increases the network overhead. In mRPL, the Trickle parameters are set according to the RPL scenario with lowest overhead ($< 12, 8 >$). The additional control messages triggered by the hand-off are invoked on-demand. Hence, in a high data rate network, similar to our example (with 30 pkts/sec), mRPL has a higher amount of overhead compared with RPL. Comparing different network topologies shows that adding more neighbor nodes (APs) increases the overhead of the network — see Figures 6.3(d), 6.4(d) and 6.5(d). Adding more APs in the neighborhood of a MN would increase the number of reply packets in the *Discovery Phase*, eventually increasing the overhead.

mRPL is very responsive to network dynamics. The total number of DAOs is an indicator for showing the efforts for creating new connections. Since RPL does not have an explicit hand-off mechanism, a successful parent selection is identified by DAO transmissions. In Topology 1 (with two APs), mRPL has the greatest number of new connections, which shows an accurate hand-off during each trip. Adding more APs in Topology 2 results in creating more connections in both RPL and mRPL. In a denser deployment (Topology 3), there are more overlaps between links and hence the total number of DAOs reduces in both RPL and mRPL. However, mRPL is still able to smoothly switch between APs with only 1.4% less hand-offs, while RPL reduces new connections up to 63% — see Figures 6.3(e), 6.4(e) and 6.5(e).

6.4.3 Further evaluations on speed, duty cycling and network density

At this stage, we aim at studying the impact of mobile node speed and network duty cycling on the performance in high and low data traffic in a more complicated network deployment. Humans tend to walk at speeds from nearly 0 m/s to upwards of 2 m/s. In our simulations, we considered a wider range of speeds from 0.5 to 4 m/s. We also assumed various data transmission periods of 50 ms, 100 ms, 500 ms, 1 s, 2 s, and 5 s. We employed a MN and 12 APs located in four rank levels (R1, R2, R3 and R4) as depicted in Figure 6.6(a). MN starts its trip from the vicinity of AP_1 and travels all the

network through the dotted lines, then pausing for 30 seconds at the initial position, while the simulation runs for two minutes.

mRPL is efficient for the range of normal human walking speed. We define efficiency when achieving fast, light (low overhead) and reliable hand-off. The results in Figures 6.6(b), (c) and (d) show that the hand-off delay and network overhead are very low, in all scenarios. There are fluctuations in packet delivery ratio. In high traffic scenarios, by increasing mobile node speed, the packet delivery drops. For instance, with 50 ms and 100 ms data periods, the packet delivery ratio drops by $\approx 4\%$ and $\approx 6\%$ respectively, when increasing MN speed from 0.5 m/s up to 4 m/s. The main reason is the change in the starting and finishing moments of the hand-off process. Some data packets in higher speed scenarios may drop due to these timing behaviors. By lowering the data transmission period, the Trickle timer and the mRPL timers reduce their periodicity to minimize network overhead. This effect causes some packet drops due to the delayed hand-off process. With low data periods, the packet delivery ratio with different mobile node speeds has more fluctuations. Successful packet delivery depends on performing a hand-off before reaching an AP. The hand-off starting moment depends on the timeouts of the timers (T_C and T_{MD}). Tuning these timers to longer periods in low activity networks causes sudden packet drops. Lack of sufficient control messages in low traffic scenarios postpones the hand-off process. By lowering the data transmission period from 100 ms to 5 s, the average packet delivery ratio drops by 24%.

mRPL has less overhead in low traffic networks. In mRPL, mobility detection is according to the link degradation (ARSSI), connectivity timer and mobility detection timer. We adapt the connectivity timer according to the larger Trickle interval and the mobility detection timer according to the data periods to reduce the network overhead. Fixed and low intervals of these timers in low traffic scenarios reduce network overhead. For instance, the overhead in a low data traffic scenario (e.g. transmitting data every 5 s) is 30% less than the high data traffic scenario (e.g. transmitting data every 50 ms).

Low traffic scenarios require data retransmissions to keep network reliability. By enlarging the mobility detection timer in low traffic scenarios, some of the data packets may drop. Upon data packet losses, the hand-off process resumes and leads to parent

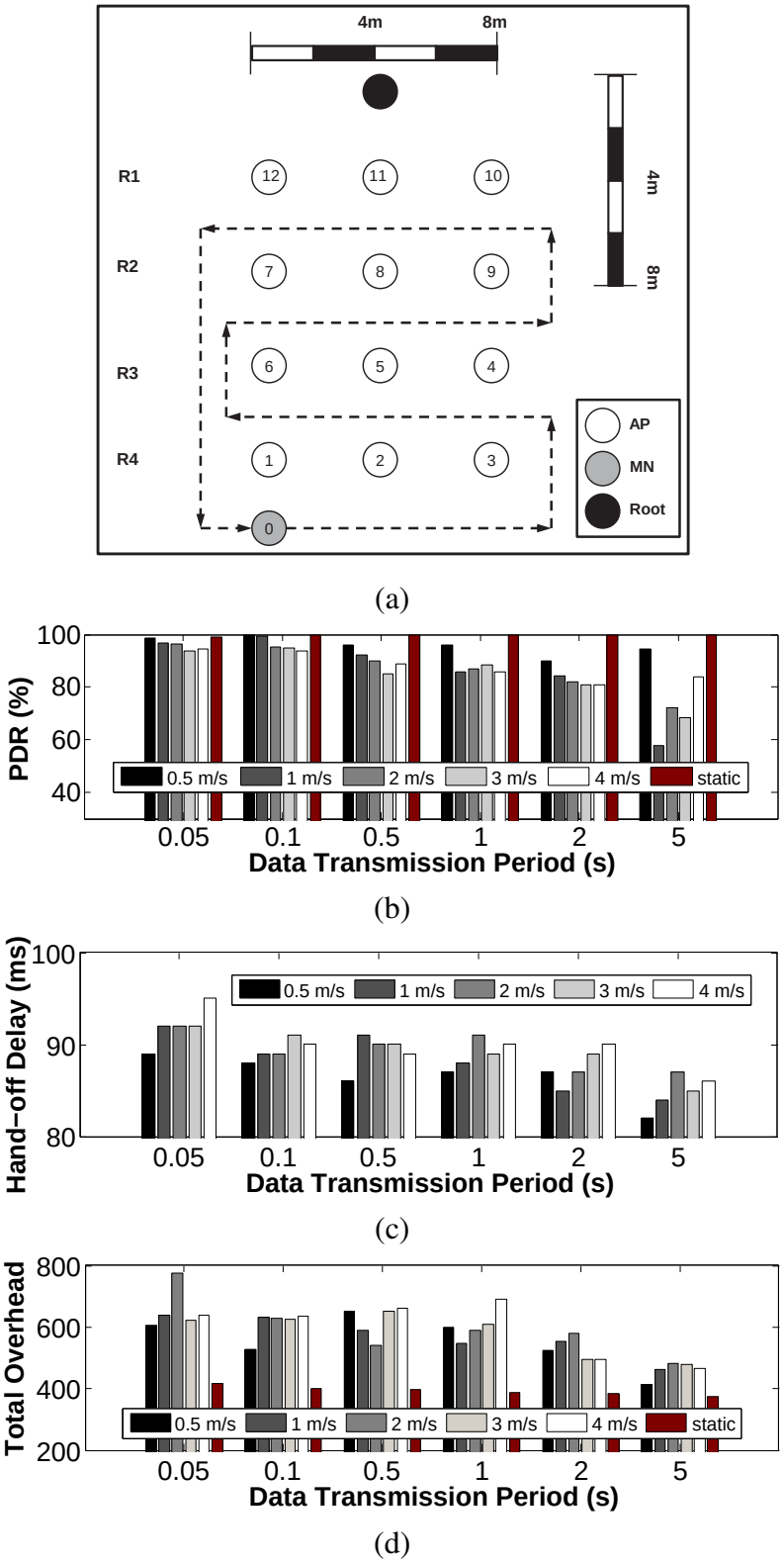


Figure 6.6: Impact of MN speed on mRPL performance with different network traffics: (a) node deployment, and performance in terms of (b) packet delivery ratio, (c) average hand-off delay, and (d) total packet overhead.

switch. After a hand-off process, a MN has good connectivity with the preferred parent. Therefore, we propose a data retransmission to the new AP immediately after the hand-off process, to keep network reliability.

Hand-off delay is low regardless of network traffic and mobile node speed. Hand-off in mRPL is a process that requires a number of packet exchanges to assess neighbor APs. This process is very fast and takes about 85 ms (with some fluctuations in various scenarios).

We also evaluated mRPL without mobile nodes. Figures 6.6(b) and (d) show that **a static node is able to successfully transmit almost all data packets to the fixed infrastructure.** The overhead of this experiment is the minimum, since additional control messages are not generated in a static environment.

The duty cycling MAC design (ContikiMAC) reduces the energy consumption by periodic idle-listen periods. In ContikiMAC, if a packet transmission is detected during a wake-up period, the radio is kept on to receive the packet. After successfully receiving a packet, the receiver sends a link layer acknowledgment. According to this behavior, in mRPL, the MN keeps sending bursts of DIS messages until receiving a reply from the neighbor AP as depicted in in Figure 6.7 (a). The radio of Receiver 1 is always on (NullMAC) and can immediately detect the packet transmissions from the MN, and thus, the hand-off process is the shortest possible. By applying a duty cycling approach, the request packets are detected later and the hand-off process takes longer (the MN keeps sending bursts of DIS messages until receiving a reply from a neighbor AP). Increasing the sleeping period worsens the performance in terms of responsiveness (compare Receiver 2 to Receiver 3)[4].

Increasing the listening period degrades the hand-off performance. We have analyzed the duty cycling approach by changing the channel check rates (64, 32 and 8 Hz) and studied the network performance — see Figures 6.7 (b)- (d). Reducing the check rates increases the listening periods, enlarging the hand-off delay. By increasing the check rate

[4]In ContikiMAC, it is required to obey a precise timing between transmissions. It uses *Clear Channel Assessment* (CCA) that reads the RSSI measurement to detect channel activity. The timing analysis in [Dun11] shows that a minimum packet size of 23 bytes is required for the CCA mechanism to work properly. We respect this limitation in our simulations and experiments as the size of IPv6-based packets is normally much larger.

from 64 Hz or 15.625 ms to 8 Hz or 125 ms (87.5% increase in listening period) with 1 (s) data transmission period, the hand-off delay increases from 115 ms to 156 ms (i.e. 26% increase), which is reasonable. Consequently, long hand-offs reduce the packet delivery ratio and increase the control message overhead. The trend of network performance degradation by increasing the listening period is similar in all scenarios (with various data transmission periods).

Hand-off delay is low in a random mobility pattern scenario. We created a scenario where the mobile node speed intermittently changes and the trajectory varies, while the mobile node moves within the deployment area. Comparing the random scenario with the constant speed scenario (2 m/s) with the 64 Hz check rate in Figures 6.6, we can see that the hand-off delay for all data traffics is below 100 ms. However, there are fluctuations in the PDR and the overhead of the random scenario compared with the constant speed scenario. The results indicate a PDR difference between 1.5% and 54% and an overhead difference from 0.2% to 14.2%. Generally, we observe that in a random speed scenario the performance for higher data traffics degrades more than for lower data traffic, when compared with the constant speed scenario.

Network density has a direct impact on the network overhead. Increasing the number of APs in a single broadcast domain increases the number of DIO replies in the *Discovery Phase* of a hand-off process, as depicted in Figure 6.8. This also increases the network overhead of mRPL. A wise deployment of APs in a real deployment, thus, reduces the network overhead drastically.

6.5 Experimental evaluation

In this section, we describe the experimental network setup in order to test and compare RPL and mRPL. The parameters' setting, topological configuration and the scenarios are described. Then, we discuss the most important results deriving from the experimental evaluation.

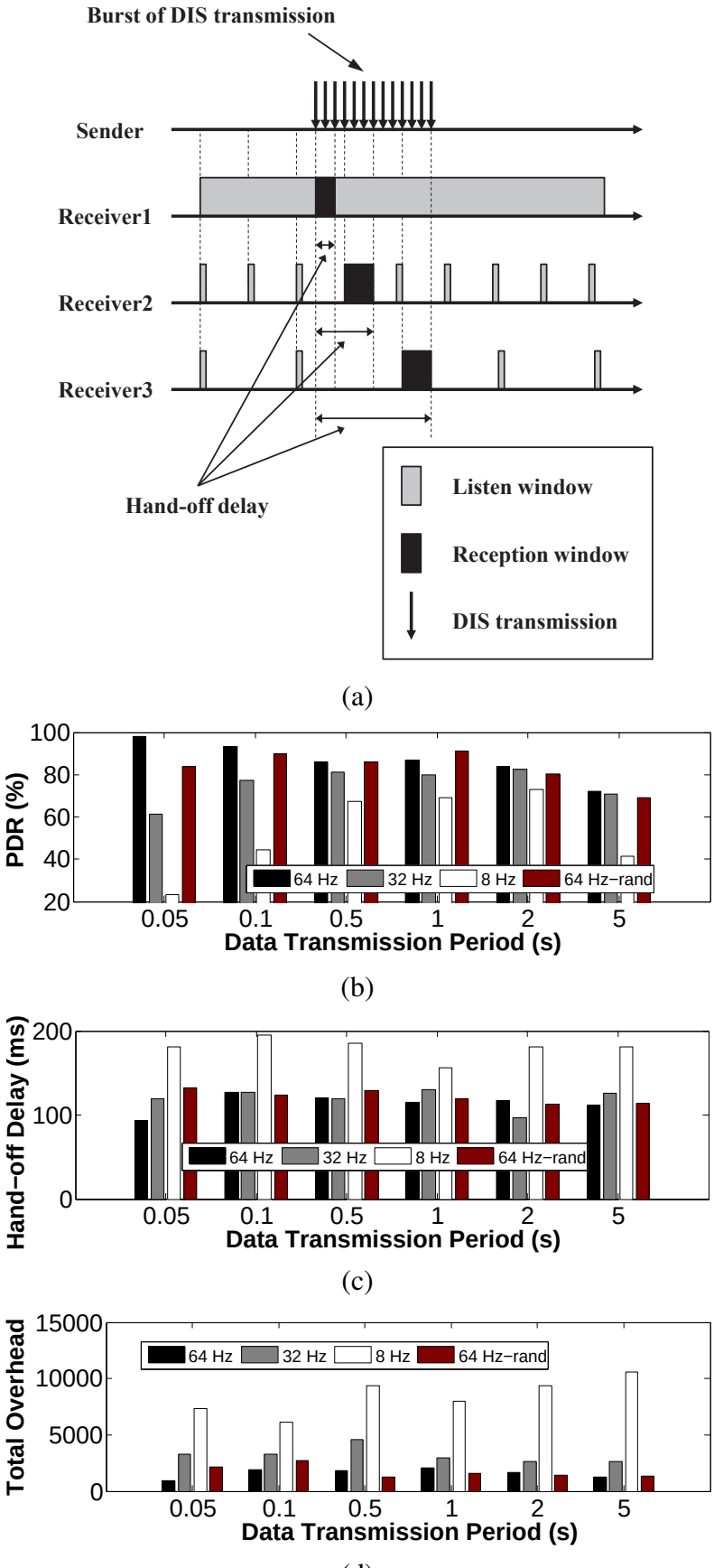


Figure 6.7: Impact of network duty cycling on mRPL performance with different network traffics: (a) an example of hand-off delay with different duty cycles, and performance in terms of (b) packet delivery ratio, (c) average hand-off delay, and (d) total packet overhead.

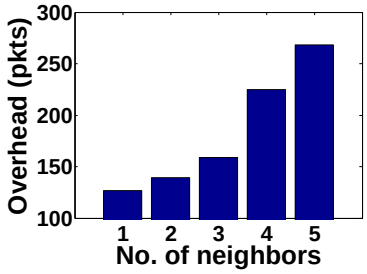


Figure 6.8: Impact of network density on total network overhead in terms of number of control message exchanges in mRPL.

6.5.1 Experimental setup

In order to perform realistic experiments, we attached the mobile node to a person's body (Figure 6.9(a)) and connected to the logging PC[5] to collect the information. The experiments were held in a big room 80 m^2 size. Figure 6.9(a) (Setup 1) shows a scenario where contiguous APs provide minimal overlap. This situation was achieved by selecting the lowest transmission power (-25 dBm) and locating APs with a 0.3 m separation. In a more realistic scenario, Setup 2, we randomly deployed 9 APs in the room (as depicted in Figure 6.9(b)). The APs were attached to walls at 1.5 m height from the ground (to guarantee a better connectivity).

RPL configurations. In general, RPL devices play the role of either a router or a root node. In our experiments, we consider a single root that collects all data. The access points and the mobile nodes are routers; the MNs generate data and the APs forward them to the root.

In order to compare RPL with mRPL, we created the best possible RPL setting to switch fast between parents when a child node moves. Typically, in RPL, a child node needs to detect a high ETX value to trigger a parent switch. The frequency of ETX updates depends on the network traffic in terms of rate of data/control exchanges. We considered the highest possible data rate to increase the RPL routing responsiveness to network dynamics.

[5] At the beginning, we connected all APs to one laptop with passive USB cables and USB2.0 hubs. Then we observed some data loss during data transfer through the UART port. Adding more PCs did not solve the problem completely. Hence, we managed to get the data log from the MN with the cost of a person carrying a laptop during the experiment.

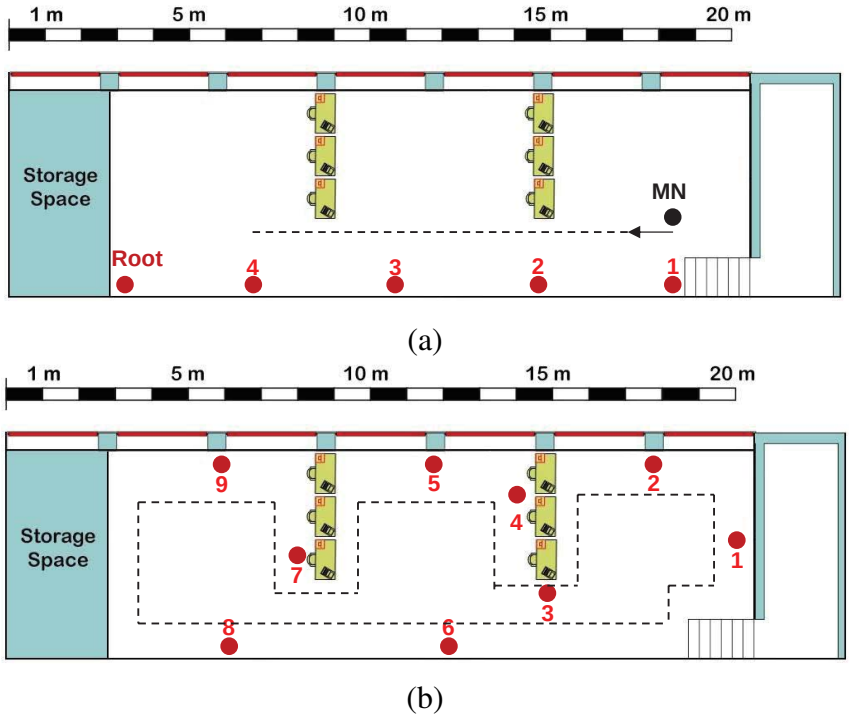
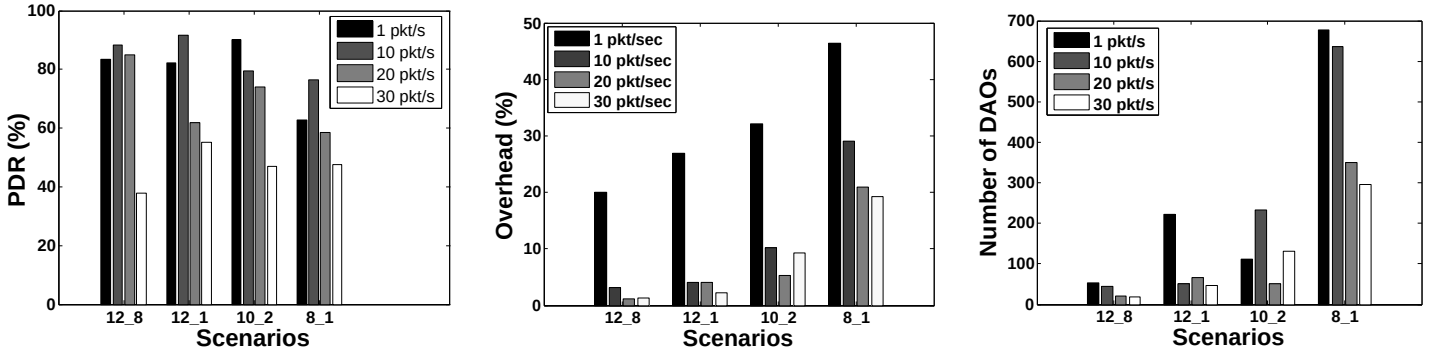


Figure 6.9: Experimental evaluation of mRPL, (a) Experimental Setup 1 with 4 APs and a MN deployed in a row, and (b) Experimental Setup 2 with 9 APs distributed across the lab.

6.5.2 Results and discussion

Experimental Setup 1. We compare various RPL scenarios with mRPL in a simple network topology presented in Figure 6.9(a), which provides minimum overlap between contiguous APs. All nodes run NullMAC (full-time on), which is more useful for comparing mRPL and mRPL without the effect of packet losses and inherent delays in a duty cycling protocol.

First, we evaluate the packet delivery ratio of various RPL scenarios (previously defined in Table 6.3) with different data rates. Our analysis indicates that **higher data traffic leads to lower packet delivery ratio.** Smaller values of the Trickle timer and high data generation rate increase the network traffic, which in turn increase the chance of packet collisions — see Figure 6.10(a). Packet drops are more significant for larger Trickle timer values (e.g. $< 12, 8 >$), which results in nearly 46% packet drops when increasing the data rate from 1 to 30 pkt/s, while the lowest timer setting ($< 8, 1 >$) exhibits around 29% drops.



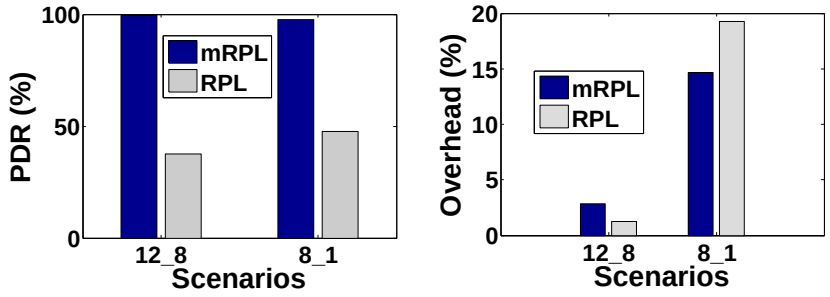
(a) (b) (c)

Figure 6.10: Experimental Setup 1, comparing several RPL scenarios in terms of (a) packet delivery ratio, (b) overhead, and (c) number of DAOs.

Smaller values of the Trickle timer impose higher control packets overhead in RPL. The overhead is calculated according to the percentage of the ICMPv6 packets over the total number of packets (ICMPv6 packets + data packets). Figure 6.10(b) shows that the overhead of RPL increases when choosing smaller Trickle values ($< 8, 1 >$ scenario results in more control message exchanges compared with other scenarios). A lower data transmission rate results in a higher percentage of control messages with respect to the total number of packet exchanges.

Successful parent switching is based on the data rate and the Trickle setting. The number of DAOs corresponds to the number of new links created between a child (MN) and a neighboring parent (AP). A high data transmission rate increases the ETX value updates. Figure 6.10(c) shows that for all settings, the higher the packet rate, the lower the DAO transmissions. Additionally, smaller Trickle timer values increase the number of DAO packets.

mRPL performance is independent of the Trickle setting. In fact, mRPL uses RPL control messages as a backup mechanism. We compared mRPL with two extreme RPL scenarios ($< 12, 8 >$ and $< 8, 1 >$). Figure 6.11(a) shows that regardless of the Trickle setting, mRPL copes with correctly delivering most of the data packets (nearly 100%). Note that reducing the Trickle timers decreases the mRPL packet reception rate by only 2%, as the data packets are more prone to collide with the control packets. The use of Trickle as a backup in mRPL raises the overhead of the algorithm in terms of additional control message exchanges. Hence, in mRPL it is recommended to use a low overhead



(a) (b)

Figure 6.11: Compare RPL and mRPL in Experimental Setup 1 in terms of (a) PDR and (b) overhead.

Trickle setting (e.g. $< 12, 8 >$, as can be inferred from Figure 6.11(b)).

Experimental Setup 2. We extended the tests by deploying 9 APs as depicted in Figure 6.9(b). All nodes were tuned to transmit power level 3 (-25 dBm), which created higher overlap between the neighbor APs. A root node was placed in the center of the room. As already referred, the mobile node was attached to a person's arm and moving along the dashed path and generating packets at different rates.

A higher overlap of the wireless links increases the packet delivery ratio. Apparently, by creating more AP coverage overlapping, more packets have the possibility to reach the destination. However, RPL cannot support high transmission rates under mobility, as shown in Figure 6.12(a). Contrarily, our experiments revealed that in an extreme condition with 30 pkt/sec data rate, mRPL still forwards 99.7% of data packets.

Figure 6.12(b) shows the overhead of RPL scenarios with different data rates. The trends are similar to the Experimental Setup 1. The best RPL setting with high data rate is $< 12, 8 >$, which leads to the lowest overhead. To update the routing information, RPL benefits from the data as well as control message exchanges. With the same setting in a high data rate application (30 pkt/s), mRPL resulted in 1% additional control messages overhead.

Higher links overlaping reduces the possibility of parent switching. The number of new links is smaller than for Experimental Setup 1. By comparing the results in Figure 6.10(c) and Figure 6.12(c) in a high data rate condition, we conclude that the new links establishment (in a more realistic network topology) for $< 12, 8 >$ and $< 8, 1 >$ RPL scenarios reduces by 14% and 31%, respectively. Thus, we infer that higher links

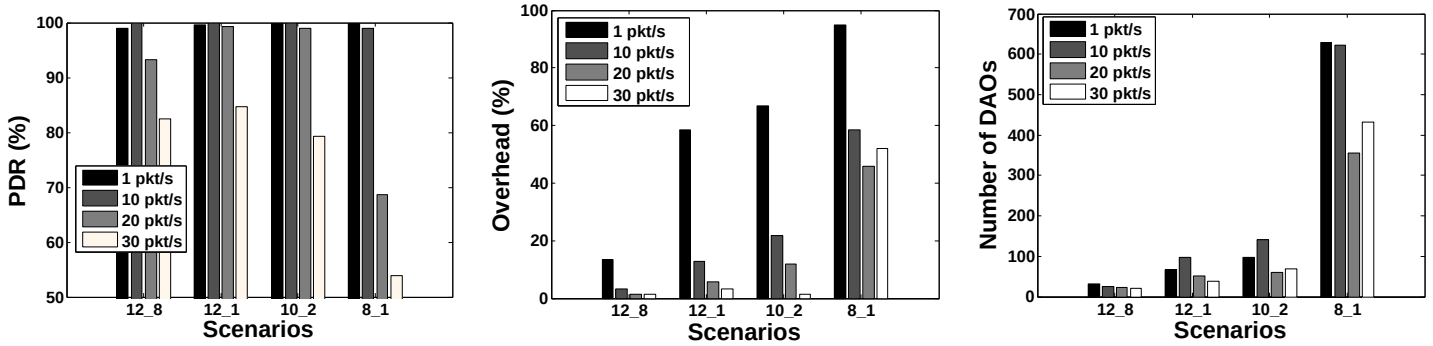


Figure 6.12: Experimental Setup 2. Comparing various RPL scenarios in a more realistic network topology in terms of (a) packet delivery ratio, (b) overhead, and (c) number of DAOs.

connectivity postpones the process of parent switching, and hence decreases the number of DAOs.

Standard RPL has no built-in hand-off mechanism. Therefore, it is hard to calculate the hand-off delay in RPL. In simulation, we have presented a rough estimation of the hand-off delay for different RPL scenarios. Empirical results show that mRPL has a very fast parent switching process with about 88 ms hand-off delay, leading to a very high packet delivery ratio even with high data transmission rates.

Further insight into Experimental Setup 2. In indoor experiments, the location of APs, furniture, people and the external interference affect the hand-off performance. Therefore, we decided to have a deeper insight on the APs activity and selection process, as follows. In Figure 6.13, the arrows correspond to the parent switching (from one AP to another). The thickness of the arrows indicates the amount of hand-off/s in the correspondent link. Note that hand-offs are not always performed between the closest APs. This means that the high variability of low-power wireless links and the dynamic behavior of the mobile network may dictate not choosing the closest APs. Figure 6.13 also illustrates (with circles) the amount of packet exchanges with mRPL at each AP. Larger circles means more packets received (Nodes 3, 5, 6 and 7). Nodes in a good connectivity region (central location) can maintain the connection longer than the ones on the right and left sides of the room (Nodes 1, 2, 8 and 9); hence more packets are successfully received by “central” APs.

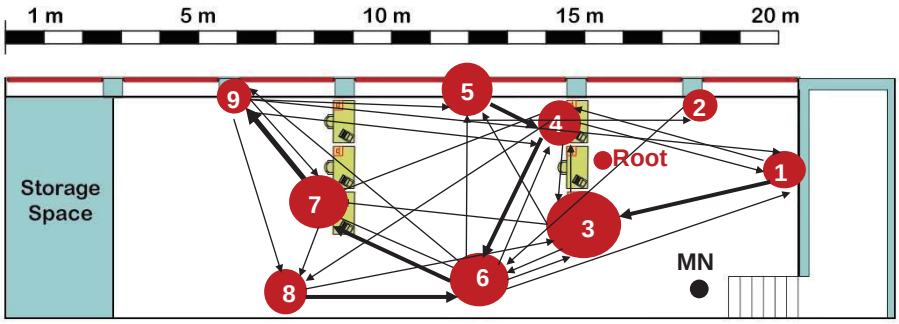


Figure 6.13: Mobile node movement representation in Experimental Setup 2.

Figure 6.14 shows the packet delivery ratio and the average RSSI at each AP (links 1 to 9 from the MN to the APs). There is a correlation between the average RSSI and the PDR in each link (higher ARSSI leads to higher PDR). This means that a hand-off triggered within the transitional region of the wireless link can result in a very good performance. Keeping the average RSSI and the PDR high requires a very careful decision on the moments of starting and finishing a hand-off: closer to the lower threshold of the transitional region would reduce the packet delivery drastically. Nodes 3 and 7 are more benefited as they are placed in more strategic places with better ARSSI.

6.5.3 Simulation vs. experimental results

We compared the results of the simulation with the experimental tests in the two network setups: (i) Experimental Setup 1 as depicted in Figure 6.9(a) and (ii) Experimental Setup 2 as depicted in Figure 6.9(b)). The results in Table 6.4 show that the performance in terms of packet delivery and hand-off delay in the experimental tests are relatively better than in the simulation. The overhead in Setup 2 reduces drastically as the links overlaping between the neighbor APs is higher than in the simulation. Moreover, the radio model of the simulator is not very accurate (mainly based on the distance between the nodes). However, in the real world, various physical and environmental parameters affect the radio model.

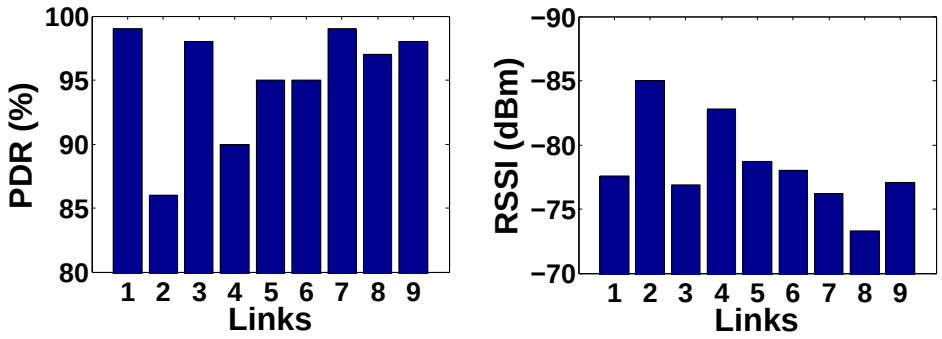


Figure 6.14: Packet delivery ratio and the average RSSI measurement of each link in Experimental Setup 2.

6.6 Conclusion

We proposed a very simple yet effective solution to cope with mobility as one of the challenging issues for future IoT applications. We extend RPL — the standard routing protocol for low-power networks based on the IoT architecture — with fast and reliable mobility support, dubbing this mobility-enabled version of RPL as mRPL.

We smoothly integrated smart-HOP within RPL in a way to keep backward compatibility with the standard protocol. The smart-HOP hand-off mechanism [FZA+12, FAZK14] was applied to mobile nodes (MNs) by managing the schedules of the control messages within the Trickle algorithm (DIS, DIO and DAO). A MN selects a new preferred parent according to the average RSSI (ARSSI). Neighboring nodes within the child set of the MN are ignored, to prevent routing loops.

We implemented some timers to increase hand-off efficiency by reducing hand-off delays and network congestion. We considered low-power link characteristics and the limitations of the IPv6 architecture to tune the schedules. We applied priorities to APs (according to the ARSSI levels) to minimize the probability of packet collisions during the hand-off process.

The integration of smart-HOP within RPL (dubbed mRPL) was tested, fine-tuned and validated through extensive simulations and experiments. The results we obtained indicate that an inaccurate radio propagation model in the simulator has some impact on the results related to the hand-off performance.

We also found that the best setting of RPL parameters for LPWNs supporting mobility

Table 6.4: mRPL: simulation versus experimental results

Scenarios	Sim. results	Exp. results
Setup 1 (Figure 6.9(b))	PDR=98.12% Overhead=18.8% Delay=81 ms	PDR=99.77% Overhead=7.63% Delay=92.7 ms
Setup 2(Figure 6.9(c))	PDR=99.56% Overhead=2.85% Delay= 86.21 ms	PDR=99.68% Overhead=2.43% Delay= 88.2 ms

leads to a huge control message exchange overhead. Instead, mRPL is able to keep a low overhead while being responsive to network changes ($\approx$ 85 ms hand-off delay). Moreover, nearly 100% packet delivery rate is achieved upon mobility.

We studied the impact of varying network traffic, duty-cycling and mobile node speed on mRPL hand-off performance. The results indicated that in low traffic networks, the hand-off process is less responsive. Moreover, enlarging the listening periods affects the performance by increasing the hand-off delay. However, the variation of mobile nodes speed (within the range of human walking speed) does not affect the overall performance.

We implemented and integrated our hand-off mechanism in the RPL/6LoWPAN stack in Contiki, a widespread operating system for low-power wireless networks. Importantly, we made the source code freely available to the international community [Pro14a].

In the next chapter, we will provide the mobility management framework for RPL routing, where it addresses the support of hard and soft hand-off mechanisms together with hand-off features and add-ons for multi-hop networks.

Chapter 7

mRPL+ design and evaluation

In this chapter, we propose a mobility management framework integrating both hard and soft hand-off approaches, dubbed mRPL+. To enable this soft hand-off mechanism with single radio nodes, we use the overhearing mechanism. We integrate both hand-off mechanisms in RPL, providing important hand-off features and add-ons (backward compatibility, collision avoidance, route establishment deferring, loop avoidance, traffic awareness and best route selection). The evaluation indicates that mRPL+ achieves a high data delivery (nearly 100%) with extremely low hand-off delay (4 ms) compared with 85 ms in mRPL and a few seconds in RPL.

7.1 Mobility management framework

In order to support mobility in multi-hop LPWN, we devise a mobility management framework that provides timeliness, reliability and efficiency, which is depicted in Figure 7.1. In this framework; we (i) employ two hand-off approaches (soft and hard hand-offs), (ii) devise mechanisms to increase hand-off efficiency, (iii) develop improvements on RPL routing to increase the end-to-end performance upon mobility (multi-hop add-ons) and (iv) extensively evaluate this mobility management mechanism by probabilistic analysis and Cooja simulation. We define the main features of the proposed mobility management framework as follows.

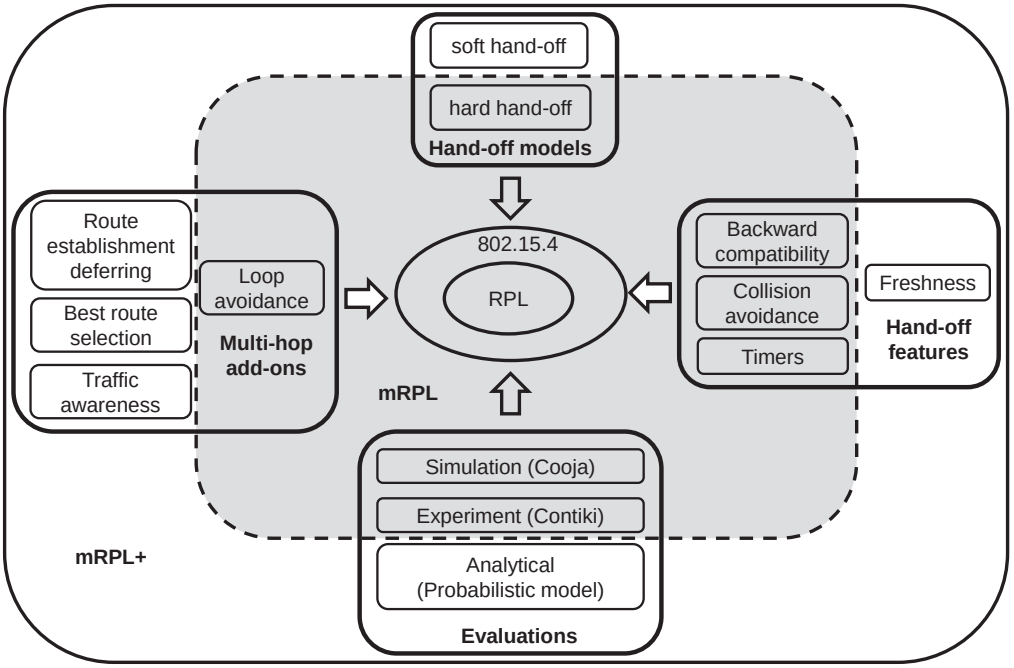


Figure 7.1: Mobility management framework in RPL routing.

– *Hand-off models.* We devise a soft hand-off model within RPL that runs on all mobile nodes and handles faster hand-offs than the hard hand-off (mRPL) that runs as a backup. Both hand-off models are integrated within RPL (dubbed mRPL+).

– *Hand-off features.* We designed and implemented algorithms to enable efficient and fast hand-offs. Backward compatibility is achieved by supporting mobile nodes without interfering with the RPL default protocol. The hand-off decision in such dynamic environment is ensured to be made very fast based on recent information (guaranteed by the freshness parameter), while the required information from neighboring APs is collected in a non-conflicting manner. Four timers are devised to increase the hand-off efficiency and detect parent unreachability and link degradation faster.

– *Multi-hop add-ons.* During a hand-off mechanism, the MN avoids choosing APs with high traffic load (*traffic awareness*) and also give higher priority to APs that have shorter distance to the destination (*best route selection*). After a hand-off mechanism, mRPL+ eliminates loop occurrence (*loop avoidance*) and avoid path breakage (*route establishment deferring*).

Through extensive simulations, we evaluate the proposed mobility management. We compare various scenarios in different situations (low/high data traffics and low/high speed of mobile nodes).

7.2 Hand-off models

As already referred in Chapter 2, there are two types of hand-off approaches in wireless networks: hard hand-offs and soft hand-offs. In a hard hand-off, the MN disconnects from the serving AP and then searches for a new AP. In a soft hand-off, MN finds a new AP before disconnecting from the serving AP. Thus, the disconnection period with a hard hand-off model is longer than with a soft hand-off model. While smart-HOP is a hard hand-off approach that has been implemented in RPL (mRPL), in this section, we present the design of mRPL+, a soft hand-off model for mRPL.

The main concept of our proposed soft hand-off model relies on overhearing mechanism, which enables continuous cooperation between a single-radio MN and neighbor APs. Each AP listens to the channel activity and continuously tracks the link quality level (e.g. ARSSI) based on packets received from the MNs. The chipcon CC2420 MAC sub-layer supports promiscuous mode (to sniff packets). By enabling the promiscuous mode, the MAC sub-layer accepts all frames received from the physical layer whether or not the node is the destination node, storing useful information for the soft hand-off mechanism in a table.

The first step in a mobility management mechanism is to detect physical mobility. For faster mobility detection, we define the following rules:

R1 The link quality degradation, which is observed by enabling the MAC sub-layer acknowledgement and measuring the estimated transmission count (ETX)[1] [DDB12] values. It means that the MN keeps reading the default RPL metric (ETX). Upon observing a low ETX value, MN stops data communication and resumes the hand-off process.

[1]RPL can use different routing metrics that define the cost of each path. We employ the most widely and well-acknowledged ETX metric to construct and maintain RPL routes. It determines the total number of transmissions needed to reach the destination. ETX is obtained by summing the inverse of the packet reception ratios in each hop in the path.

R2 Receiving a reply from serving AP with $ARSSI < T_\ell$ triggers/starts a hand-off mechanism.

R3 Expiration of the parent unreachability timer upon detecting inactivity in the link indicates node mobility.

In our mobility management framework, we utilize both hand-off models (hard and soft). The soft hand-off runs as a default and the hard hand-off is called in any of the following conditions:

C1 Receiving no reply from the current AP within a pre-designed time frame (e.g. 100 ms).

C2 Receiving no reply from neighbor APs before disconnecting from the serving AP (e.g. 500 ms before hand-off).

C3 Observing link quality degradation based on either ARSSI or ETX metrics, and receiving no reply from neighbor APs during the last 500 ms.

C4 Expiring the parent unreachability timer and receiving no reply from neighbor APs.

Figure 7.2 depicts an example of mRPL+ with soft and hard hand-offs. By enabling the overhearing mechanism, neighbor APs can sniff the communication between the MN and the serving AP. In this model, unlike mRPL (hard hand-off mechanism), the neighbor APs are proactive by probing the MN. The neighbor AP with good quality link with the MN ($RSSI > T_h$) sends a beacon to inform its willingness for being the future parent AP. In this example, we assume that the MN is initially connected to AP_1, while the serving AP and the neighbor APs (within the communication range of MN) receive the data and control packets transmitted from the MN.

In the soft hand-off model, the current AP replies if and only if the ARSSI of a number of consecutive packets received from the MN is smaller than the lower threshold level (T_ℓ). In the meanwhile, the neighbor APs overhear the messages and measure the link quality; after each data transmission, they measure the RSSI. The neighbor AP that measures ARSSI above a certain threshold (T_h) sends a DIO to the MN, holding the ARSSI

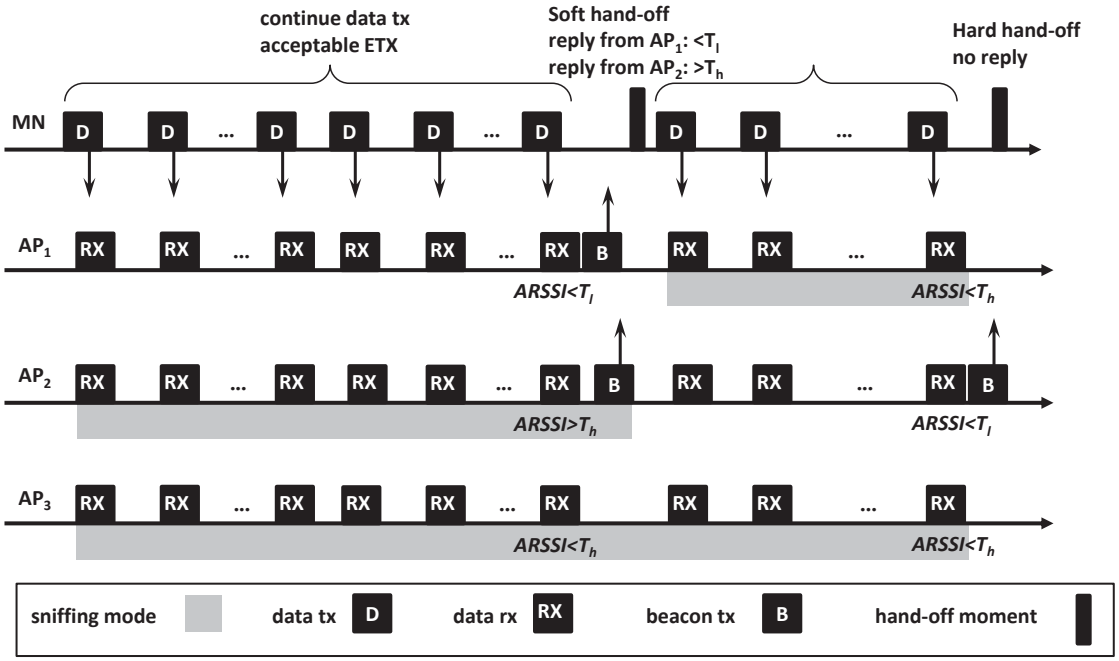


Figure 7.2: An example of mRPL+ with three APs and a MN. The x-axis illustrates the order of the processes in mRPL+, considering data/control message exchanges and hand-off mechanism.

measurement and the address of the AP (ID or IP). If the MN detects mobility based on one of the aforementioned rules (e.g. receiving a message with $ARSSI < T_\ell$ from the current AP, say AP_1), and receives a message from a neighbor AP with $ARSSI > T_h$, say AP_2, then the MN quickly switches from AP_1 to AP_2.

Consider the case where the MN detects node mobility (in this example, ARSSI of AP_2 goes below T_ℓ), and it has no message received from the neighbor APs concerning their willingness for future service, then the MN resumes a hard hand-off process. In this case, MN broadcasts burst of DIS messages to neighbor APs to get replies from high quality links (Discovery Phase). This phase continues until finding an AP with good link quality. MN simultaneously delivers data to the last preferred AP during the Discovery Phase, to reduce the possibility of packet loss.

7.3 Multi-hop add-ons

Mobility support within a routing protocol requires careful considerations. In a hand-off decision, the MN should consider network condition during and after hand-off. Hence, we consider network parameters such as local traffic and rank level within a DODAG. In the following subsections, we will provide details on multi-hop add-ons during and after a hand-off. The reference multi-hop routing protocol for our mobility management framework is RPL.

7.3.1 Multi-hop add-ons during a hand-off

In order to select the best AP in a multi-hop network, we consider both link and network metrics. The link quality degradation triggers the hand-off process. Then, the combination of link and network parameters are used for selecting the best AP. The hand-off process implicitly considers the network parameters according to the following mechanisms.

Traffic awareness. A MN may temporarily experience a good quality link (i.e. $ARSSI > T_h$) with an AP that has a high traffic load. Choosing this AP as the next parent node increases the possibility of packet drops and would eventually incur unnecessary hand-offs. To avoid this problem, APs with high traffic load (e.g. observing activity within 50 ms time frame) turn off their promiscuous mode to disable overhearing. Consequently, APs with high traffic load do not send DIO to the MN in order to avoid being selected as the future parent.

Best route selection. The MN may require to forward packets to the root or to a destination that is another router/leaf node. The parent selection during the hand-off process affects network performance in terms of end-to-end delay. In our approach, in case of sending data to the root, a MN prefers APs with lower rank level (closer to the root). For the case of sending data to router/leaf nodes, a MN should have the information of destination path from the neighboring APs. Each AP embeds the address of the destinations in the DIO messages. This information is available in RPL routing and thus does not add any communication and computational overhead. Upon receiving DIOs from neighbor APs, a MN selects the AP that has a path toward the destination. *A path from an AP toward*

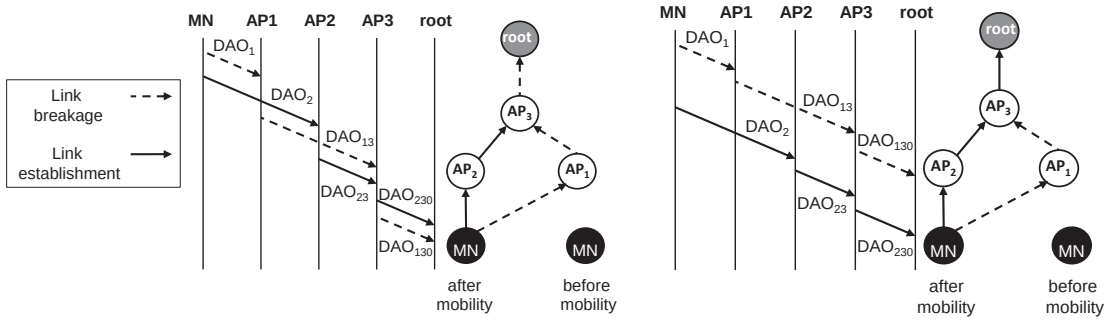


Figure 7.3: Examples to show downward path breakage: (a) default DAO transmissions in RPL after a parent switching from AP_1 to AP_2 and link breakage occurrence (MN-AP_1, AP_1-AP_3 and AP_3-root). The link breakage between AP_3 and the root disables downward data communication from the root toward the MN. (b) DAO deferring to avoid path breakage. Note a DAO_{130} means that DAO unicasted from MN to AP_1, forwarded to AP_3 and then reaching the *root* (AP_0).

a designated destination exists if and only if the DIO message received at the MN holds the destination address. If the selected AP has no path toward the destination, packets are forwarded to the root (the AP with lowest rank).

7.3.2 Multi-hop add-ons after a hand-off

The process of parent switching during hand-off affects the routing tree. We devise two approaches for maintaining downward routes and avoiding loops after mobility in RPL, which are described below.

Route establishment deferring. After hand-off, downward routes may break. This problem affects the downward data communication from the destination toward the MN. The two-way communication is beneficial in many applications. For instance, in clinical monitoring, where the source nodes are patients and the destination nodes are nurses/doctors, the destination node may request a measurement from the source node.

Downward routes are established based on unicasting DAO control messages after a parent switch. First, the MN transmits a DAO message to the previous AP to disconnect the previous downward path (from destination to the AP). This message travels toward the destination node to disconnect all the path. Then, the MN sends another DAO message to the new AP to create a new path (from the root node to the new serving AP).

Figure 7.3(a) shows an example of route establishment in RPL after a parent switching when a MN travels from the vicinity of AP_1 toward AP_2. RPL routing unicats a DAO message to AP_1 and then immediately another DAO message to AP_2. The DAO transmission is forwarded by intermediate APs until reaching the destination. In some cases, there are longer communication or processing delay that would postpone the DAO transmission. Hence, a shared link between AP_1 and AP_2 may first forward the DAO for establishing a downward link and then forward the DAO for disconnecting the downward link (link between AP_3 and the root, in this example). This problem causes path breakage, discarding all packets that are sent from the root to the source node.

To reduce the possibility of downward link breakage, we delay the DAO transmission upon establishing a new downward path. Figure 7.3(b) shows the case where the DAO transmission to AP_2 is delayed to increase the chance of downward link establishment toward the new serving parent.

Loop avoidance. This process was previously implemented for mRPL. The type of mobile node (leaf or router) affects the loop avoidance strategy. For mobile leaf nodes, there is no possibility of loop occurrence after a parent switch, because the mobile node not a router. However, in case of mobile routers, there is a possibility of loop occurrence. During a hand-off process, the MN may connect to one of its child nodes that has a good link quality. To avoid this problem, **children decline to reply to their parents' requests during hand-off.** This simple approach reduces the network overhead and avoids loops in RPL.

7.4 Hand-off features

In order to provide efficient and low-overhead hand-off, we devised two main features: (i) *backward compatibility* with RPL, and (ii) *collision avoidance*. These features that are also employed in mRPL are summarized below.

Backward compatibility. This process enables all RPL nodes to transmit control messages without interfering with the Trickle timer interval. In order to guarantee the coexistence of default RPL nodes and mobility-enabled RPL nodes, we enhance RPL con-

trol messages instead of using new messages. We add few bits to DIS and DIO messages to: (i) distinguish between the normal RPL control messages and the hand-off control messages, and (ii) carry the link quality value (ARSSI) and the list of destinations that have a path to the AP for the *best route selection*.

Collision avoidance. This process provides a conflict free medium for packet exchanges between the MN and the neighbor APs. In order to reduce the possibility of collision, we apply two main strategies: (i) poor links ($ARSSI < T_h$) decline to notify the MN and (ii) high quality APs reply earlier than low quality APs, in predetermined time-slots. We define a new parameter (*prio*) to distinguish between low and high priority link. *prio* is set to 0 for high quality measurements ($ARSSI \geq -85$ dBm) and is set to 1 for lower quality measurements (-85 dBm $< ARSSI < -85$ dBm).

There is a possibility of measuring high quality links at more than one neighbor AP, resulting in setting $prio = 0$ at those APs. We consider an additional random delay ($rand(t_1, t_2)$) to allocate different time slots for transmitting DIO packets. The range of random delay is $t_1 = 10$ ms and $t_2 = 15$ ms, which are values above the maximum possible transmission of IP-based packets (4 ms) with the CC2420 radio.

The offset time for transmitting DIO replies follows this equation: $t_{offset} = t_2 \times prio + rand(t_1, t_2)$. In the worst case, a low quality link ($prio = 0$) waits for t_2 ms for transmitting a DIO packet to the MN ($max(prio \times t_2 + rand(t_1, t_2)) = t_2$).

In our system model, we are considering a wise deployment of APs in order to avoid very high or very low density of APs, providing a minimal overlap between contiguous APs that prevent the possibility of having multiple high quality APs in a region. In any case, the proposed collision avoidance scheme prevents concurrent transmission of DIO packets.

Freshness. It is important to take hand-off decisions based on accurate information. In a mobile network, the hand-off is triggered based on link quality information. The links in low-power and mobile networks are dynamic and change during node mobility. Thus, we keep the time-stamp of all messages from alternative APs together with their address and ARSSI measurements. By elapsing a time window (e.g. 500 ms), the information of an AP is removed from the buffer. This simple approach reduces the effect of environmental

dynamics on the link quality information.

Timers. We implemented four timers (as previously described in Chapter 6) to increase hand-off efficiency. The main tasks of these timers are: (i) to detect link degradation of serving AP and inform the MN promptly, (ii) to detect parent unreachability by requesting a reply from the serving AP upon observing an idle link (no data/control message exchange), (iii) to manage replies from neighbor APs during soft and hard hand-offs and (iv) to manage the periodicity of DIS transmissions during the Discovery Phase.

7.5 Analytical model and evaluation

Environmental characteristics may greatly affect network performance in low-power wireless networks. In this section, we extend the probabilistic analysis of smart-HOP [FAZK14] to a more general case that supports both soft and hard hand-offs. In this probabilistic analysis, we study the impact of two major channel parameters:

1. *path-loss exponent (η).* It measures the power of radio frequency signals relative to distance.

2. *standard deviation (σ).* It measures the standard deviation in RSSI measurements due to log-normal shadowing.

The values of η and σ change with the frequency of operation and the clutter and disturbance in the environment. At this stage, we study the hand-off performance in various environmental conditions to observe the feasibility and efficiency of the algorithm. It is important to consider a "probabilistic" or "analytical" model that is faithful to the underlying physical model while being amenable to analysis. We assume a scenario consisting of two APs (AP_a and AP_b) that are 6m apart and a MN that moves from the vicinity of AP_a toward AP_b.

The two main hand-off performance metrics are the probability of ping-pong effect and the expected hand-off delay. We show the hand-off occurrence of soft and hard hand-off models in Figures 7.4(a) and (b). The MN moves with a constant speed of 1 m/s. In a hard hand-off model, the MN disconnects from AP_a when the RSSI value reaches

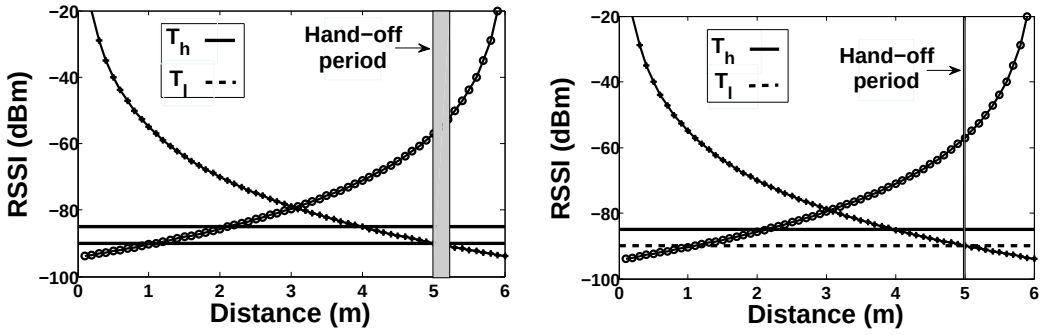


Figure 7.4: Hand-off system models in case of (a) hard hand-off, and (b) soft hand-off. The threshold levels are assumed -85 and -90 dBm. The hand-off happens in the shadow areas.

the lower threshold level T_ℓ. From this point onward, the MN stops communication with AP_a and tracks the RSSI of the neighboring AP, AP_b. If the mobile node observes a signal strength above a higher threshold level, T_h, the hand-off process is considered to be finished. The hand-off period is marked with a shadowed vertical bar in Figure 7.4(a). However, in a soft hand-off process, the MN receives RSSI updates from the neighbor AP while sending data to AP_a. By observing a high link quality ($RSSI > T_h$), the MN switches from AP_a to AP_b. The soft hand-off process is very fast as depicted in Figures 7.4(b).

Link quality monitoring. There are different ways of measuring the link quality metric. In this work, we consider the RSSI as the link quality level. The reason is that in low-power networks with unreliable links, and using hardwares with weak processing capabilities (e.g. telosB) to sense, sophisticated link quality metrics are not responsive to environmental changes. Hence, we selected RSSI parameter, which is simply measured by the antenna.

The probabilities of being below the lower threshold level and above the higher threshold level are defined by using a Q-function. The traveling path of the MN is divided into a number of slots. For the sake of simplicity, we consider the same sampling rate for both the *Discovery* and *Data Transmission Phases*. These probabilities are expressed as follows.

$$P(R_a(i) < T_\ell) \stackrel{\text{def}}{=} Q(\frac{-T_\ell + R_a(i)}{\sigma}) \qquad P(R_b(i) > T_h) \stackrel{\text{def}}{=} Q(\frac{T_h - R_b(i)}{\sigma})$$

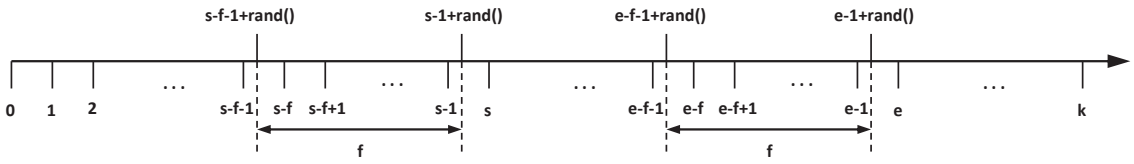


Figure 7.5: Moments of starting and ending the hand-off process.

Where $Q(.)$ is the complementary distribution function of the standard Gaussian, i.e., $Q(x) = \int_x^\infty (1/\sqrt{2\pi})e^{-t^2/2}dt$, $R_a(i)$ and $R_b(i)$ indicates the RSSI values from AP_a and AP_b at slot i, and σ (in dB) expresses the standard deviation.

Radio model. The received signal strength is estimated by a log-normal shadowing path-loss. According to this model, $R(i)$ (in dBm) (RSSI level at a given slot i) from the transmitter is given by [Rap96]:

$$R(i) = P_t - \overline{PL(d_0)} - 10nlog_{10}(i/d_0) - X_\sigma \tag{7.1}$$

Where i corresponds to distance, P_t is the transmission power, $\overline{PL(d_0)}$ is the measured path-loss at reference distance d_0, n is the path-loss exponent, and $X_\sigma = N(0,\sigma)$ is a Normal variable (in dB). The term X_σ models the path-loss variation across all locations at distance i from the source due to shadowing, a term that encompasses signal strength variations due to the characteristics of the environment (i.e., occlusions, reflections, etc.).

Hand-off probabilistic model. In this model, we combine both hard and soft hand-off processes. It means that in case of not receiving a high link quality information from a neighbor AP, the MN enters a hard hand-off process. To evaluate the performance metrics, we define the possibility of starting and ending a hand-off process at each slot. Figure 7.5 shows an example where the MN encounters disconnections and connections in different time slots. The probability of starting a hard hand-off process at slot $s \in [1,k)$ is defined as follows (k indicates the total number of slots).

$$P_{\text{hard}}(S(s)) = \left[\prod_{i=1}^{s-1} P(R_a(i) \geq T_\ell)\right] \times P(R_a(s) < T_\ell) \tag{7.2}$$

The first part of the equation indicates the observation of a number of slots $(s-1)$ with good/acceptable link quality level (above T_ℓ). The second part denotes the observation of

the low link quality for the first time (below T_ℓ). The following settings are used in all future evaluations across this section: $\sigma = 4$ dB, $\eta = 4$, $P_t = 0$ dBm, $d_0 = 1$ m, $d = 6$ m, $PL(d_0) = -55$ dB, $m = 1$, $T_\ell = -90$ dBm and $T_h = -85$ dBm.

In a soft hand-off process, MN receives information from AP_b while connecting to AP_a. This information drops if it exceeds the freshness value (f). According to the definition, if MN has an alternative parent and the RSSI of the current link drops below T_ℓ, then MN switches from AP_a to AP_b. The third part of the Equation 7.3 indicates the sniffing information during the f period. We employ different slots for receiving these information in order to eliminate collision. The $rand()$ value is used to change the period between slots.

$$P_{\text{hard-soft}}(S(s)) = \left[\prod_{i=1}^{s-1} P(R_a(i) \geq T_\ell)\right] \times P(R_a(s) < T_\ell) \times \prod_{s-f-rand()}^{s-rand()} [1 - (P(R_b(j) < T_h))] \quad (7.3)$$

We compared the probability of starting a hand-off process in mRPL (hard hand-off) with mRPL+ (hard and soft hand-offs), while varying the f variable as depicted in Figures 7.6. The results show that mRPL+ performs the hand-off later with higher probabilities of occurrence within smaller time windows. We also observe that increasing the f variable (i) postpones the hand-off process and prolongs the current connection, (ii) increases the probability of starting the hand-off within a time window, and (iii) reduces the starting hand-off time window (sharpens the signal). We conclude that **a wise freshness value with mRPL+ results in a more accurate hand-off mechanism by keeping the current connection longer and performing the hand-off in a proper time window.**

In a hard hand-off process, the MN starts assessing the other neighboring AP at slot s to find a moment that the link quality exceeds the T_h. Equation 7.4 formulates the probability of ending a hand-off at slot e, considering the fact that the hand-off would have been started at slot s and the MN was disconnected from either AP_a or AP_b.

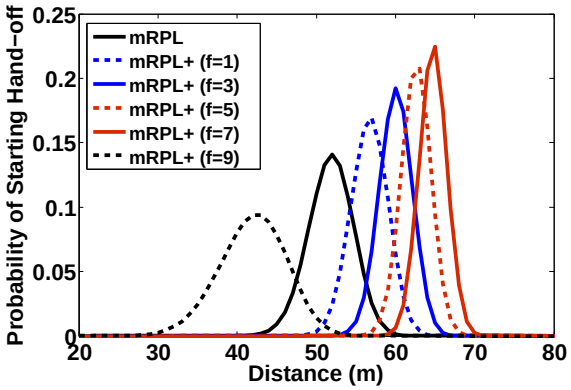


Figure 7.6: Probability of starting hand-off in mRPL+ considering different f values with 100 ms time-slot assignment.

$$P_{\text{hard}}(E(e) \mid S(s)) = \left[\prod_{i=s+1}^{e-1} (P(R_a(i) < T_h) \times P(R_b(i) < T_h)) \right] \times \left[\prod_{e-ws-1}^{e-1} (1 - (P(R_a(j) < T_h) \times P(R_b(j) < T_h))) \right] \tag{7.4}$$

This equation assumes that the hand-off occurs at slot s. The ending moment at slot $e \in (s+1, k]$ happens at a later stage by comparing the RSSI level of the APs to a higher threshold level, T_h. Hence, in practice it is a conditional probability and depends on a situation that has taken place previously at slot s.

To end up the soft hand-off process, the MN should have observed a high link quality information from either AP_a or AP_b during the period f before the ending slot at e. If the MN has an alternative parent, the ending slot is $e = s+1$, but in case of hard hand-off process it takes more slots to assess the link qualities of the neighbor AP during a ws period. Equation 7.5 involves both hand-off models, since we cannot guarantee receiving a high link quality information before disconnecting from a serving AP (soft hand-off).

$$P_{\text{hard-soft}}(E(e) \mid S(s)) = \left[\prod_{i=s+1}^{e-1} (P(R_a(i) < T_h) \times P(R_b(i) < T_h)) \right] \times \left[\prod_{j=e-f-1+rand()}^{e-1+rand()} (1 - (P(R_a(j) < T_h) \times P(R_b(j) < T_h))) \right] \tag{7.5}$$

The time span between the starting slot and the ending slot is called hand-off delay. It is possible to calculate the hand-off delay for all possible hand-offs starting at any slot. By considering each point as a starting moment of a hand-off process, we characterize the ending moments by the probabilities defined in Equations 7.4 and 7.5. The expected hand-off delay is computed by getting the weighted sum of all possible hand-off periods. It is defined as the product of the time spent in each possible hand-off process started at slot s and ended at slot e by the correspondent probabilities of starting a hand-off at slot s, $P(S(s))$, and ending it at slot e, $P(E(e) \mid S(s))$. For each hand-off starting at slot s, the hand-off would end at one of the slots from $s+1$ to k. The sum of all these possible situations defines the expected delay for a hand-off started at a specific slot s. The overall expected hand-off delay is defined as follows.

$$\text{Delay}(s,e) = \sum_{s=1}^{k-1} \sum_{e=s+1}^{k} ((e-s) \times P(E(e) \mid S(s)) \times P(S(s))) \tag{7.6}$$

We compared the hand-off delay in mRPL and mRPL+ based on Equation 7.6 as depicted in Figure 7.7. The results show that the disconnection period in mRPL+ is less than mRPL implementation due to the possibility of supporting soft hand-off process within mRPL+. Increasing the freshness value f reduces the hand-off delay and then it stays unchanged at some point. In this example $f = 11$ results in 100 ms hand-off delay, which is the least possible disconnection period. This means that **a proper freshness setting results in an efficient mRPL+ mechanism, which enables a soft hand-off process and a fast parent switching.**

In order to measure the ping-pong effect, a new term is defined that is called probability of restarting a hand-off. This situation happens when a MN performs hand-off at an improper moment, thus leading to an unnecessary hand-off. The restarting of a hand-off always occurs after successfully ending the first hand-off at slot $r \in (2,k]$. This means that the probability of restarting is also a conditional probability that depends on ending a hand-off at an earlier stage. Since the MN may have been connected to either AP_a or AP_b, the signal strength should be evaluated for both cases. The equations for the hard hand-off and integrated hard and soft hand-off are defined as follows.

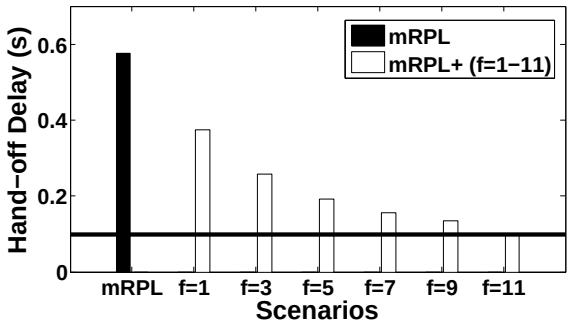


Figure 7.7: Comparing mRPL with various mRPL+ scenarios by varying f variable in terms of hand-off delay. The horizontal line depicts the minimum possible hand-off delay that takes within one time-slot (100 ms).

$$P_{\text{hard}}(R(r) \mid E(e)) = \left[\prod_{i=e+1}^{r-1} (1 - P(R_a(i) < T_\ell) \times P(R_b(i) < T_\ell)) \right] \times [P(R_a(i) < T_\ell) \times P(R_b(i) < T_\ell)] \tag{7.7}$$

$$P_{\text{hard-soft}}(R(r) \mid E(e)) = \left[\prod_{i=e+1}^{r-ws-1} (1 - P(R_a(i) < T_\ell) \times P(R_b(i) < T_\ell)) \right] \times [P(R_a(j) < T_\ell) \times P(R_b(j) < T_\ell)] \times \left[\prod_{j=r-f-rand()}^{r-rand()} (1 - P(R_a(i) < T_\ell) \times P(R_b(i) < T_\ell)) \right] \tag{7.8}$$

The second disconnection of the MN in this trip will be ended at slot p, which is defined in the following equation.

$$P_{\text{hard}}(P(p) \mid R(r)) = \left[\prod_{i=r+1}^{p-1} (P(R_a(i) < T_h) \times P(R_b(i) < T_h)) \right] \times \left[\prod_{i=p-ws+1}^{p} (1 - (P(R_a(i) < T_h) \times P(R_b(i) < T_h))) \right] \tag{7.9}$$

$$P_{\text{hard-soft}}(P(p) \mid R(r)) = \left[\prod_{i=r+1}^{p-1} (P(R_a(i) < T_h) \times P(R_b(i) < T_h))\right] \times \left[\prod_{j=p-f-1-rand()}^{p-1+rand()} (1 - (P(R_a(j) < T_h) \times P(R_b(j) < T_h)))\right] \tag{7.10}$$

To find out the probability of ping-pong effect, the full history of a MN since the first start of hand-off to the end and restarting and ending are taken into account. Equations 7.11 and 7.12 illustrate the cases that lead to a ping-pong effect at slot p, as follows.

$$\begin{aligned} P_{\text{hard}}(P(p)) = {} & \left[\left(\prod_{i=1}^{s-1} P(R_a(i) < T_\ell)\right) \times P(R_a(s) < T_\ell)\right] \times \left[\prod_{i=s+1}^{e-1} (P(R_a(i) < T_h) \times P(R_b(i) < T_h))\right] \\ & \times \left[\prod_{i=e-ws-1}^{e-1} (1 - (P(R_a(j) < T_h) \times P(R_b(j) < T_h)))\right] \times \left[\prod_{i=e+1}^{r-1} (P(R_a(i) < T_h) \times P(R_b(i) < T_h))\right] \\ & \times [(1 - (P(R_a(i) < T_h) \times P(R_b(i) < T_h)))] \times \left[\prod_{i=r+1}^{p-1} (P(R_a(i) < T_h) \times P(R_b(i) < T_h))\right] \\ & \times \left[\prod_{j=p-ws-1}^{p-1} (1 - (P(R_a(i) < T_h) \times P(R_b(i) < T_h)))\right] \end{aligned} \tag{7.11}$$

$$\begin{aligned} P_{\text{hard-soft}}(P(p)) = {} & \left[\prod_{i=1}^{s-1} P(R_a(i) \geq T_\ell)\right] \times P(R_a(s) < T_\ell) \times \left[\prod_{s-f-rand()}^{s-rand()} 1 - (P(R_b(j) < T_h))\right] \\ & \times \left[\prod_{i=s+1}^{e-1} (P(R_a(i) < T_h) \times P(R_b(i) < T_h))\right] \times \left[\prod_{j=e-f-1-rand()}^{e-1-rand()} (1 - (P(R_a(j) < T_h) \times P(R_b(j) < T_h)))\right] \\ & \times \left[\prod_{i=e+1}^{r-1} (1 - P(R_a(i) < T_\ell) \times P(R_b(i) < T_\ell))\right] \times [P(R_a(j) < T_\ell) \times P(R_b(j) < T_\ell)] \\ & \times \left[\prod_{i=r+1}^{p-1} (P(R_a(i) < T_h) \times P(R_b(i) < T_h))\right] \times \left[\prod_{i=p-f-1-rand()}^{p-1-rand()} (1 - (P(R_a(j) < T_h) \times P(R_b(j) < T_h)))\right] \end{aligned} \tag{7.12}$$

We define total probability of ping-pong effect in a more abstract way as appears in Equation 7.13. For each case of hand-off occurrence at slot $s \in (0, k-2]$, there is a chance to finish the hand-off at one of the upcoming slots $e \in (s, k-1]$. Similarly for each e, as an ending slot, there is a chance of restarting another hand-off at slot $r \in (e, k]$. Thus:

$$\text{Total probability of ping-pong} \stackrel{\text{def}}{=} \sum_{s}^{k-3} \sum_{e}^{k-2} \sum_{r}^{k-1} \sum_{p}^{k} P(S(s)) \times P(E(e) \mid S(s))$$

$$\times P(R(r) \mid E(e)) \times P(P(p) \mid R(r)) \tag{7.13}$$

7.6 Simulation evaluations

In this section, we explain the simulation set-up in order to test and compare the mRPL+ with mRPL and with the default RPL. The parameters' setting, topological configuration and the simulation scenario are described. We implemented and tested mRPL+ in the Cooja simulator as it easily ports to the sensor hardware and provides the opportunity to analyze different network conditions.

Performance metrics. The evaluation focuses on the impact that hand-off parameters have on three network metrics:

- *Hand-off delay.* In mRPL and mRPL+, it represents the average time required to perform the hand-off process and in standard RPL routing it expresses the time spent to discover a new preferred parent in the standard RPL.

- *Total packet overhead.* We classify all the non-data packets (control messages) as network overhead. RPL uses ICMPv6 based control messages (DIS, DIO and DAO) for building and maintaining DODAGs. The mRPL and mRPL+ utilize these control messages to detect mobility and perform the hand-off process.

- *Packet delivery ratio (PDR).* It is defined as the number of successfully received packets at all APs over the total number of packets sent from all MNs. The successful delivery rate of mRPL+ is compared with mRPL and RPL, in the presence of mobility.

Memory overhead. The memory overhead of the standard RPL against mRPL and mRPL+ is illustrated in Table 7.1. mRPL adds 8.4% (3.7 kB) memory (ROM) on the AP nodes, while mRPL+ adds 13.3% (6.2 kB) memory. These values increases to 9.3%

Table 7.1: Memory usage in RPL, mRPL and mRPL+.

Implementation	ROM (bytes)	RAM (bytes)
RPL (MN)	40,202	7,660
mRPL (MN)	44,348	8,562
mRPL+ (MN)	46,820	7,988
RPL (AP)	40,336	7,606
mRPL (AP)	44,022	8,512
mRPL+ (AP)	46,514	7,926

(4.1 kB) additional memory for the MN in mRPL and 14.1% (6.6 kB) in mrPL+. We conclude that mRPL+ requires $\approx 5\%$ more memory than mRPL.

Simulation set-up. We consider a MN, 12 APs and a root node that collects the data generated by the MN, deployed in a 8 m $\times$ 20 m room size — see Figure 7.8. APs are deployed in such as way to create multi-hop behavior (five rank levels). The MN starts its trip from the vicinity of AP_1 and travels all the network through the dotted lines, then pausing for 30 seconds at the initial position, while the simulation runs for two minutes. The speed of the MN is limited to the range of human walk speed: from 0.5 m/s to a 2 m/s.

We consider two extreme RPL scenarios ($< 12, 8 >$ with DIO intervals ranging from 4.096 s to 1048.576 s, and $< 8, 1 >$ with DIO intervals ranging from 0.256 s to 0.512 s), which are determined based on Trickle parameters (I_{min} and $I_{doubling}$). We consider different data transmission periods, from very high traffic to a very low traffic generation rates (50 ms, 100 ms, 500 ms, 1 s, 2 s, and 5 s). We also consider different mobile node speeds (0.5 m/s, 1 m/s, and 2 m/s) and study the impact of speed on hand-off performance. In the following subsection, we describe the performance evaluation of mRPL+ compared with mRPL and RPL.

7.6.1 mRPL+ vs mRPL and RPL

We run several simulations by considering different data generation rates from 0.05 ms to 5 s with normal human walk speed (v = 1 m/s). Figure 7.9(a) depicts the hand-off delay of RPL with scenarios $< 12, 8 >$ and $< 8, 1 >$. In a stable network topology, the Trickle timer enlarges the DIO intervals. Thus, mobility detection is slower after infrequent control

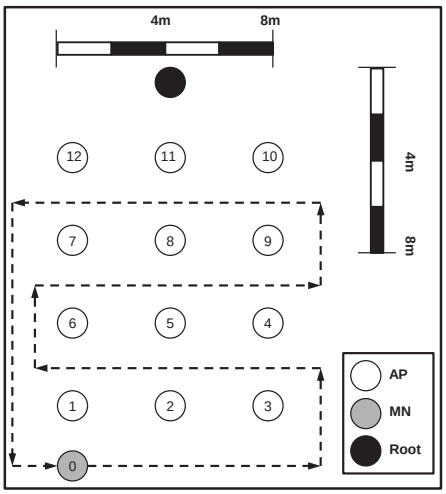
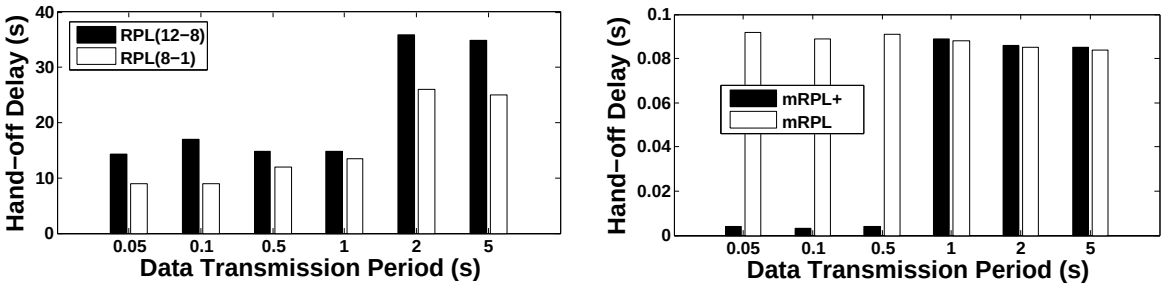


Figure 7.8: Node deployment.

message exchanges, which leads to network inaccessibility and enlarges hand-off delays. In RPL with smaller Trickle intervals (e.g. $< 8, 1 >$ in this example), the hand-off delay is less as RPL is able to assess links more often by using lower Trickle intervals.

mRPL+ performs hand-off in few milliseconds. In mRPL, the hand-off delay is about 85 ms, and in mRPL+ (featuring the soft hand-off mechanism), it drops to about 4 ms. Figure 7.9(b) illustrates that increasing the data transmission period does not affect the hand-off delay. Thus, a MN is always able to find a new parent and switch between APs in a very short time.

In RPL, by reducing network traffic, network overhead (in terms of number of control messages) reduces. The average overhead of larger interval scenario is $\approx$ 92% higher than the lower interval scenario. The number of data packets in a 2-minute simulation reduces by $\approx$ 83% (from 144 packets with 50 ms data period to 24 packets with 5 s data period). However, the total overhead reduces 4.3% in RPL with $< 8, 1 >$ scenario and 11.65% in RPL with $< 12, 8 >$ scenario — see Figure 7.10(a). This means that the majority of packet exchanges in RPL are control messages (that maintain network connectivity). In order to adjust the rate of control message exchanges, the Trickle timer reduces the periodicity of control message exchanges in lower traffic conditions. We conclude that RPL with larger interval setting has extremely low overhead. Hence, we employ RPL with $< 12, 8 >$ in our mRPL and mRPL+ evaluations.



(a) (b)

Figure 7.9: Hand-off delay with different data transmission periods,(a) comparing RPL scenarios, and (b) comparing mRPL with mRPL+.

By adapting mobility timers according to the traffic load, the total overhead reduces in lower traffic scenarios. Hand-off-enabled RPL models (mRPL and mRPL+) depict very low overhead, which is in the same range as the most low-overhead RPL scenario ($< 12, 8 >$). In mRPL and mRPL+, the mobility detection timers (T_C and T_{MD}) are tuned according to the network traffic, in order to reduce mobility assessments in lower traffic scenarios. We observe a decreasing trend in total overhead while the data transmission period increases. In high traffic scenarios (e.g. in 50 ms, 100 ms and 500 ms in this example), mRPL+ detects node mobility based on the overhearing mechanism. The results in Figure 7.10(b) show that in these situations, the network overhead of mRPL+ is slightly smaller than in mRPL. The soft hand-off mechanism avoids resuming the hard hand-off mechanism. However, for lower traffic loads (e.g. in 1 s, 2 s and 5 s in this example), mRPL+ has slightly higher overhead. The reason is that the mobility detection based on overhearing in low traffic networks is not sufficient. Then, the hard hand-off process is called to assess the neighbor APs, by multicasting bursts of DIS messages.

The packet delivery ratio in RPL fluctuates for different traffic loads. In RPL, successful data delivery depends on the successful route maintenance during parent switching. RPL considers static nodes, where the link quality between parents rarely changes. In a stable network, mobility detection and successful packet delivery depends on the DIO intervals. In some cases, node mobility is detected in a right moment and in others not. Thus, successful packet delivery fluctuates in different data traffic scenarios as depicted in Figure 7.11(a).

Hand-off-enabled RPL models ensure network reliability. In both mRPL and

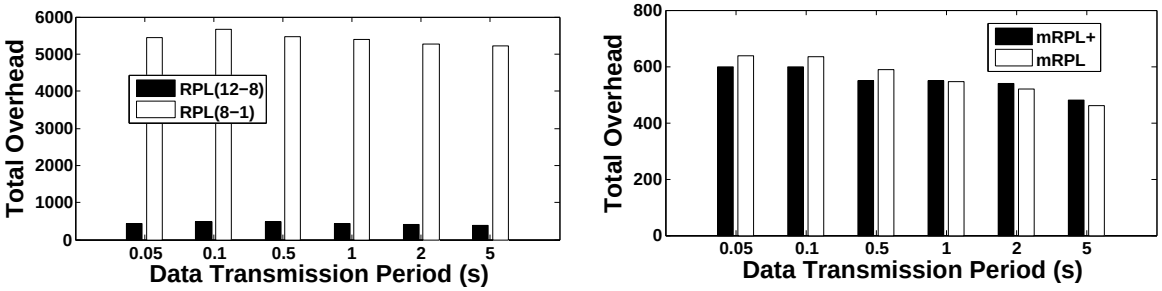


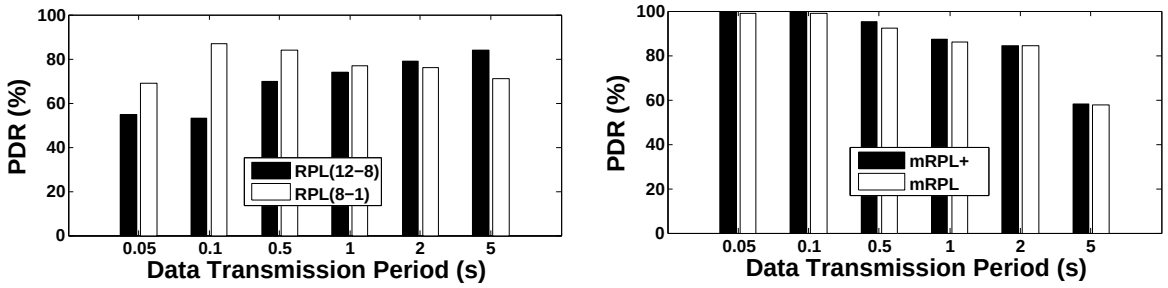
Figure 7.10: Total network overhead with different data transmission periods,(a) comparing RPL scenarios, and (b) comparing mRPL with mRPL+.

mRPL+ models, the packet delivery ratio is $\approx 100\%$ in high traffic scenarios — see Figure 7.11(b). The reason is that the number of control/data packet exchanges permit the algorithm to detect node mobility in the right moments. However, by intentional reduction in the number of control messages in low traffic scenarios, the hand-off process may be delayed and eventually detecting node mobility becomes slower; a late hand-off process affects successful data delivery. However, we believe that re-transmitting the dropped packets is more efficient than keeping network overhead higher.

We reduce network overhead by 27% when data transmission period increases from 50 ms to 5 s, which is equal to 178 less control packets (transmitting 461 control packets in low traffic scenario). By reducing network traffic, the packet delivery ratio drops from 100% (delivering all 7200 data packets) to 72% (delivering 84 data packets out of 120 packets) in mRPL+. It means that in low-traffic networks, by avoiding transmitting 178 control packets and making the network less responsive to detect nodes mobility, it results in 36 data packet drops. Educated guess would be to unicast dropped packets (36 data packets) instead of incurring in high network overhead by multicasting (178 control packets).

7.6.2 Impact of nodes speed

In this subsection, we study impact of mobile nodes speed by assuming speeds 0.5, 1 and 2 m/s for different traffic loads. In general, increasing node speed in RPL routing increases network disconnectivity, which leads to larger hand-off delays (as depicted in



(a) (b)

Figure 7.11: Packet delivery ratio with different data transmission periods,(a) comparing RPL scenarios, and (b) comparing mRPL with mRPL+.

Figure 7.12(a)). However, there are some fluctuations in scenarios with different speeds that is mainly due to the sudden Trickle updates. In mRPL, hand-off delay is roughly constant ($\approx$ 85 ms) with different speeds and network traffics — see Figure 7.12(b). With an efficient hard hand-off mechanism (smart-HOP), the parent switching process takes about one *Discovery Phase* to perform a successful hand-off. The functionality of mRPL+ with different speeds is quite similar to mRPL (Figure 7.12(c)). However, the range of hand-off delay in mRPL+ varies from 3 ms in high traffic scenarios to 85 ms in low traffic scenarios. The reason is due to the use of mobility detection timers that are able to identify environmental dynamics. Hence, we conclude that **node speeds in the range of human walk does not affect the hand-off delay.**

Figure 7.13(a) illustrates that in RPL with different node speeds, packet delivery ratio fluctuates. The reason is that mobility detection is a random process, based on the moments when the Trickle timer is triggered. However, in mRPL and mRPL+, higher nodes speed slightly reduces packet delivery ratio (Figure 7.13(b) and (c)). With higher node speed, mobility timers are not dynamically changing their intervals. Thus, the hand-off-enabled RPL models are less responsive in higher node speeds and some data packets drop due to delayed hand-off process. An estimation of node speed would enable to accurately adjust mobility timers. However, to have an accurate estimate, a localization algorithm should be employed, imposing additional overhead and processing (we assume no information on nodes location).

The network overhead also depicts fluctuations — see Figure 7.14(a). In mRPL and mRPL+, increasing node speed slightly increases network overhead (Figure 7.14(b) and

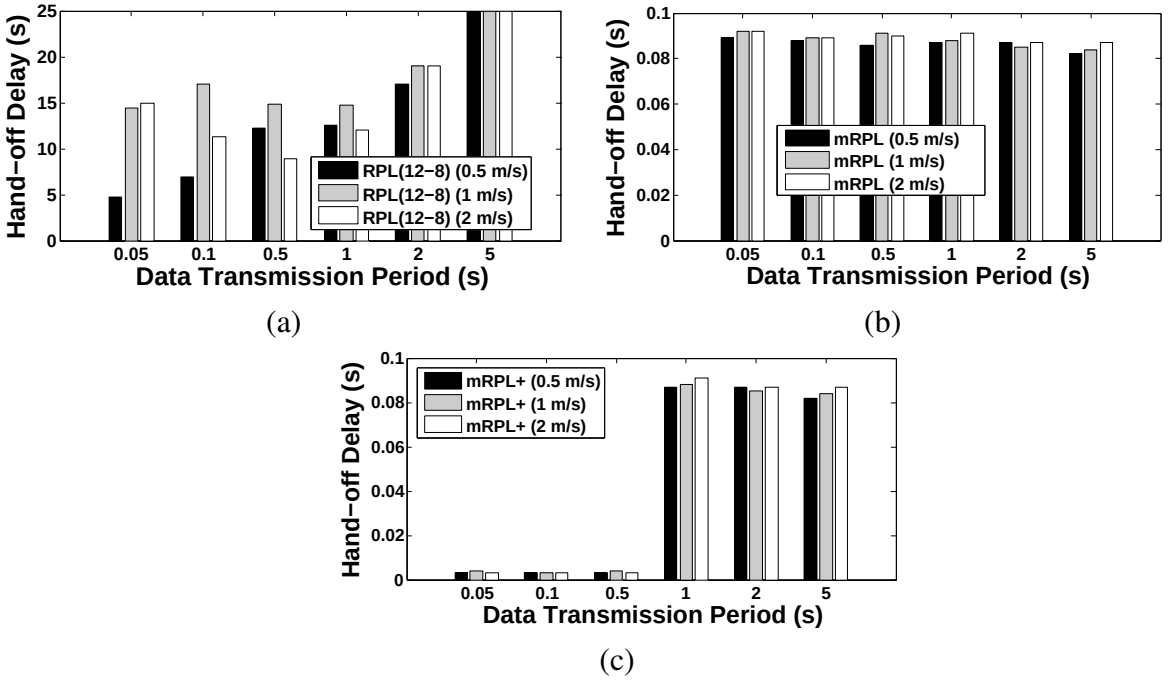


Figure 7.12: Hand-off delay with different node speeds in (a) RPL routing, (b) mRPL and (c) mRPL+.

(c)). The reason is that the human walk speed affects the moments of hand-offs. The hand-off process requires exchanging a limited number of control messages. Changing hand-off moments with different nodes speed would impose additional control message exchanges, in order to trigger a successful hand-off that will consequently increase network overhead.

7.7 Conclusion

In this section, we proposed a mobility management framework embedding hard and soft hand-off models. In the soft hand-off model, the neighbor APs employ an overhearing mechanism to observe the link activity with the MNs. By detecting a high quality link, a beacon is transmitted by the AP to the MN, stating its adequateness willingness to serve as the future parent. We analyzed the performance of mRPL (hard hand-off within RPL) and mRPL+ (soft and hard hand-offs within RPL) through extensive simulation-based evaluation. The results revealed that in high traffic networks a soft hand-off model is able to provide good reliability ($\approx 100\%$ packet delivery ratio) with extremely low hand-off delay

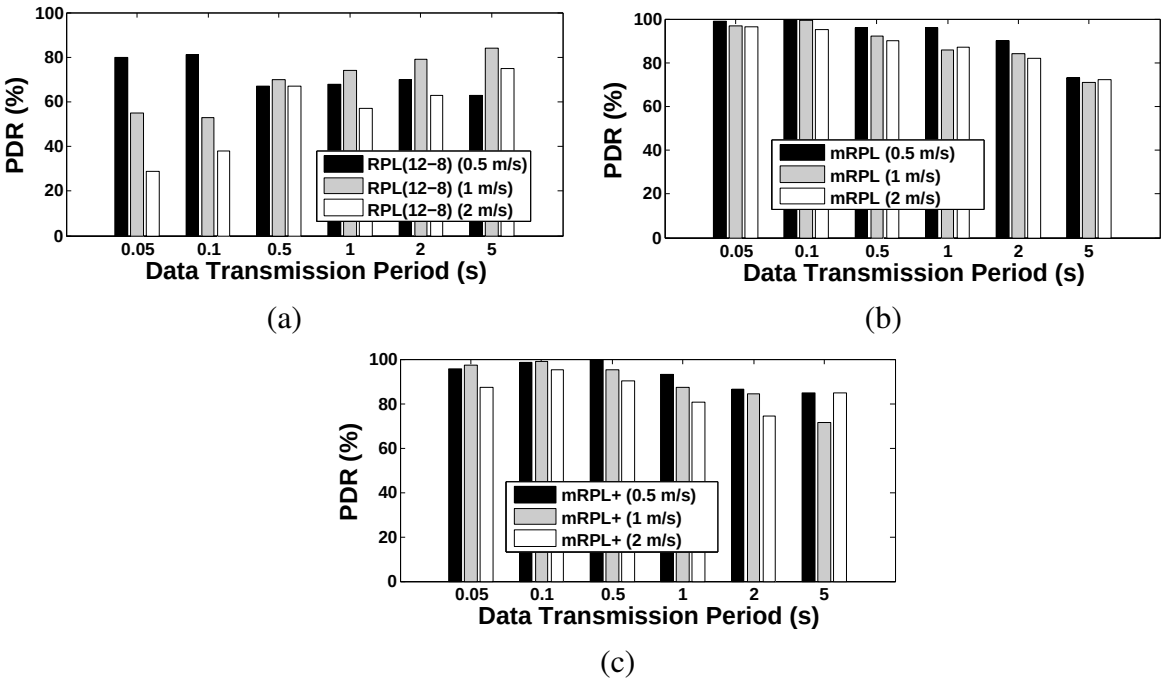
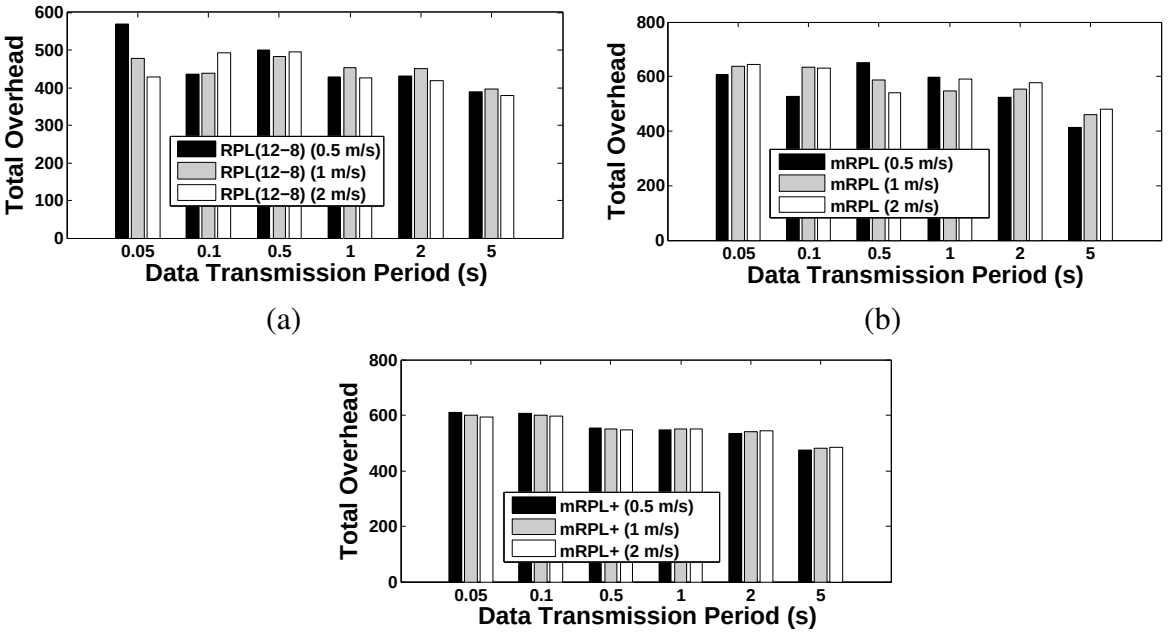


Figure 7.13: Packet delivery ratio with different node speeds in (a) RPL routing, (b) mRPL and (c) mRPL+.

(4 ms) and very low overhead (similar to RPL). In low traffic networks, mRPL+ outperforms RPL in network accessibility. However, since we adapt mobility timers according to the data transmission period (to keep the network overhead low), the mobility detection mechanism becomes less responsive; due to the lack of a localization algorithm, timers are not called in precise moments. We also found that the re-transmission of dropped packets is more efficient than exploiting additional control messages (by lowering the mobility timers).

In order to increase hand-off efficiency, we developed features such as (i) backward compatibility, to easily integrate within RPL routing without adding extra control messages, (ii) timers, to detect mobility and manage hand-off timing, (iii) collision avoidance, to reduce the possibility of control packet collision during hand-off, and (iv) freshness, to keep the most updated link and network information when deciding for parent switching.

In order to maintain multi-hop efficiency, we developed the following add-ons: (1) traffic awareness, to filter out the high traffic neighbor APs, (2) best route selection, to select the AP that has the shortest path toward the destination, (3) route establishment, to maintain downward routes from the root toward the mobile node, and (4) loop avoidance,



(a)

(b)

(c)

Figure 7.14: Total network overhead with different node speeds in (a) RPL routing, (b) mRPL and (c) mRPL+.

to avoid loop occurrence after a hand-off process.

The next chapter wraps-up the Thesis and gives further research directions.

Chapter 8

General conclusions and future directions

This chapter reviews the research objectives of the Thesis and summaries its major results, highlighting how the research contributions fulfilled the original research objectives. Some guidelines for future research work in the area of mobility support in low-power wireless networks are also provided.

8.1 Summary of the work

The need for mobility support in low-power wireless networks (LPWNs) is widely increasing for application scenarios, such as industrial automation, health-care monitoring, process control and smart cities. In this Thesis, we concentrated on providing hand-off mechanisms that consider the limitations of unreliable links in LPWNs and fine-tuned the relevant hand-off parameters. We also developed multi-hop hand-off support for a COTS routing protocol — RPL. In the following subsections, we will review the research objectives and the main contributions in this context.

8.1.1 Review of research objectives

To address the problem of mobility support in LPWNs that provides high reliability and timeliness, we defined the following three theoretical objectives:

1. Devising efficient hand-off mechanisms for LPWN considering the limitations of unreliable links (single radio nodes, links variability, and low data rate),

2. Integrating the proposed hand-off mechanisms within SOTA multi-hop networks for LPWNs (i.e. RPL routing),

3. Evaluating the mechanisms through analytical, simulation and experimental models, to show the efficiency of the proposed mobility solution.

8.1.2 Main research contributions

This Thesis provides the following contributions towards mobility support in LPWNs.

Evaluating link quality estimators (LQEs) for mobility support. Radio link quality estimation is one of the challenging issues in LPWNs. Accurate LQEs require many packet exchanges and processing capabilities. To this end, we devised a fuzzy logic-based hand-off mechanism [FAKB10], which was built on top of F-LQE [BKY+10]. In this work, we considered several parameters, such as energy level, traffic load, and routing depth. After initial experimental tests, we found that such complex heuristic-based hand-off mechanism is inefficient for low data rate and low processing devices. Hence, we opted for using single link quality metrics (RSSI and SNR). Through the Thesis, to increase the accuracy and compensate sudden fluctuations, we used averaged values.

Hard hand-off design and evaluation. We designed a hard hand-off mechanism (dubbed smart-HOP) [FZA+12, FAZK14]. We considered hand-off parameters such as low/high *threshold levels (T_l, T_h)*, *hysteresis margin (HM)*, *stability monitoring (m)* and *link monitoring (ws)*. We calibrated these parameters by considering the main features of the transitional region in LPWNs. We implemented smart-HOP in TinyOS over COTS platforms (i.e. TelosB and AdvanticSYS).

In this context, we developed a probabilistic model to study the impact of channel parameters: *path loss exponent* and *shadowing standard deviation*. We also devised a simulation model to verify the analytical model and further study the impact of hand-off parameters such as stability monitoring and link monitoring. We fine-tuned the main hand-off parameters (T_l, HM, m and ws) through extensive experiments conducted in

controlled and realistic environments, so that smart-HOP eliminates the ping-pong effect. Experimental results revealed that with the following parameter setting, smart-HOP achieves hand-off delays of a few tens of milliseconds and relative packet delivery ratios (relative to a broadcast scheme) around 98%: (i) RSSI low threshold level closer to the lower bound of the transitional region $T_l = -90$ dBm, (ii) hysteresis margin $HM = 5$ dBm, (iii) stability monitoring $m = 1$, and (iv) link monitoring $ws = 3$.

Integration of hard hand-off within RPL/6LoWPAN. To support mobility in multi-hop network scenarios, we integrated smart-HOP within a commodity protocol for LP-WNs — RPL/6LoWPAN [FMA14b]. This integration was carried out in a backward-compatible way, i.e. to enable the coexistence of standard RPL nodes and mobility-enabled nodes. In mRPL, RPL control messages were used to detect mobility and perform hand-offs. Some timers were implemented to increase hand-off efficiency (reducing hand-off delay and control message overhead). We considered the limitations of low-power links and the IPv6 architecture to tune these timers. mRPL was tested, fine tuned and validated through extensive simulation-based and experimental evaluation, using the Cooja simulator and Contiki operating system.

We found that even the best RPL parameter setting in mobile environments leads to huge control message exchanges and network inaccessibility times. Instead, mRPL is able to keep a low overhead, while being responsive to network changes (hand-off delay $\approx$ 85 ms) and delivering most data packets ($\approx$ 100%). We studied the impact of network traffic, duty cycling and mobile node speed on the overall hand-off performance. In low traffic scenarios (data transmission period greater than 1 s), the hand-off process is less responsive: enlarging the listening periods affects the performance by increasing the hand-off delay. However, mobile node speed in the range of human walk did not lead to any performance degradation (against the native RPL protocol).

Soft hand-off mechanism and integration within RPL. We developed a soft hand-off mechanism that uses the overhearing mechanism [FMA15]). We designed this hand-off mechanism, mRPL+, in a way to perform seamless hand-offs with no disconnection time (from disconnecting one access point and searching for a new one). The candidate AP informs the mobile node about its availability before the mobile node disconnects

from the serving AP. We integrated the soft hand-off mechanism within RPL, while the hard hand-off mechanism may be called as a backup.

The results revealed that in high traffic networks, the soft hand-off model provides high reliability ($\approx$ 100% packet delivery ratio) with extremely low hand-off delay (4 ms), and very low overhead. In low traffic networks, mRPL+ outperforms mRPL in network connectivity. By adapting the mobility timer according to the data transmission rate, in low traffic conditions, the mobility detection becomes slower. We found that re-transmitting dropped packets immediately after the hand-off process is more efficient than keeping fixed mobility timers with small intervals (that increase the network overhead).

We developed some hand-off features to increase a fast and efficient switching process: (1) *backward compatibility*, to enable the co-existence of both RPL and mRPL+ nodes, (ii) *timers*, to provide mobility detection and timely hand-off process, (iii) *collision avoidance*, to prohibit colliding control message exchanges (during hand-off process) and (iv) *freshness*, to keep the most recent information for hand-off decision. We also developed some multi-hop add-ons to increase multi-hop efficiency during and after a hand-off, which are: (i) *traffic awareness*, to filter out the high traffic neighbor APs, (ii) *best route selection*, to select APs with the shortest path to the destination, (iii) *route establishment deferring*, to maintain downward routes after a hand-off, and (iv) *loop avoidance*, to eliminate loop occurrence after a parent switch.

8.1.3 Validation of the hypothesis

We have shown through extensive analytical, simulation, and experimental evaluation that the proposed mobility management framework enables reliable and timely communications in standard/COTS-based LPWNs upon node mobility, adding negligible overhead and no extra hardware.

8.2 Future directions

Within the context of mobility support in LPWNs, we identify the following research directions:

Localization technique to increase hand-off efficiency. In indoor environments, the mobility pattern is limited by the structural boundaries. Estimating mobile node location and its moving trajectory helps in devising more accurate hand-off mechanisms. So, it would be possible for the mobile node to compute its location and vector speed in the Discovery Phase. The best AP is selected based on the link quality level of the AP and the predicted value for the future distances of the mobile node to the corresponding AP. This method is based on the assumption of knowing the location of fixed APs. The MN is able to compute its location after getting location information from neighbor APs, which is embedded in the reply packets (response to beacons in the Discovery Phase).

Accurate and fast link quality estimator. In this Thesis, we employed single link quality parameters, which are simply measured by the radio (RSSI and SNR). Many LQEs have been designed for LPWNs that consider more parameters and employ heuristic models. We believe that more powerful devices such as Shimmer [BGM+10], Egs [KWS+10], Opal [JKK+11] and Iris [Inc14] would enable using more accurate LQEs. Moreover, devices with additional radios are able to manage hand-off process in parallel with the normal data communication. This would eliminate the Discovery Phase during a hard hand-off and the overhearing mechanism during a soft hand-off.

Application specific mobility solution. The hand-off approaches in this Thesis were proposed for applications that support fixed node infrastructure, human walk mobility model and source node mobility. The application requirements may change in other domains that affect the design of mobility management. The lack of fixed node infrastructure, more complex mobility patterns, mobile sinks, more strict time guarantees for real-time support, higher degree of node mobility, more energy constraints, data aggregation and security issues are some examples that were not in the scope of this Thesis. A mobility management framework may require selecting additional parameters, tuning parameters, more powerful sensor hardware and additional radios.

Optimizing RPL objective function for mobile applications. In mobile routing protocols with quality of service (QoS) requirements, multiple routing metrics are essential. Routing algorithms that determine routes based on a single metric like number of hops, remaining battery power or link quality between intermediate nodes are no longer suffi-

cient, especially for networks with mobility support. RPL is a distance-vector protocol for IPv6 networks that maintains Directed Acyclic Graph (DAG) topologies toward root nodes. The topologies are built proactively according to an objective function. Since RPL must be able to operate in multiple types of deployments, it is flexible regarding the rules to form topologies and to select next hops for individual packets. We argue that a multi-objective function that considers both link and networks parameters is more efficient for mobile networks. For instance, the F-LQE/RM [BKYA14] that adapts the F-LQE (a fuzzy-based link quality estimator) within CTP (collection tree protocol) [FGJ+06] improved the CTP routing performance significantly.

Appendix A

TinyOS implementation

This appendix outlines the implementation of the smart-HOP mechanism in TinyOS.

A.1 MAC scheme

smart-HOP runs a simple TDMA-based MAC on the APs in order to avoid collisions and reduce the hand-off delay. Each AP performs a simple modulo operation on its unique *id* to obtain a specific time-slot. The *id* of APs is in the range of $[1,9] \times n$ where $n \in N$. By performing a modulo operation with value 10, the *id* of APs becomes in the range of $[1,9]$, which takes at most 9 slots. The MN is assigned an *id* which is a multiple of 10, e.g. 0, 10, 20 and etc. In theory, two nodes could collide, for example APs with *id*s 14 and 24 would select the same time slot 4, in practice, clock drifts and a relatively low density of access points (≤ 10) makes this unlikely.

The MAC scheme is not in the scope of our work. The main idea on choosing a simple TDMA-based MAC is to ignore packet collision. In our evaluation, the modulo operator is 10; while in the *preliminary experiments* the length of time-slot was considered 5 ms and in the *extended experiments* it is raised to 10 ms. In extended experiments, due to the higher link dynamics, higher density of APs, more link overlapping compared with the controlled experiments, and higher communication delay, we increased the size of time slots. The preliminary experiments were performed in a controlled environment with toy train, which enables repeatability of the experiment. The extended experiments were

conducted in a realistic environment, which a person held the mobile node walking in a room.

A.2 Implementation details

We implemented smart-HOP in TinyOS 2.0.2 [LMP+04] and used telosB [CT14] motes for the evaluation. In TinyOS, packet-level communication has two main classes of interfaces; send and receive. Each type of device (MN and AP) should handle these events according to their specific job in smart-HOP implementation. For instance, the MN is supposed to broadcast beacons in the Discovery Phase and then switch to unicast transmission of data in the Data Transmission Phase. On the other hand, the APs should process the packets that are received by MNs and then *reply* packets should be directed to the requested MN only with customized information. Existence of different phases on each device increases the complexity of run-time computations and processing. The algorithm should be smart enough to process and communicate, considering the limitations on memory and processing capability.

Initially, there is no association between nodes in the network. For instance, the MN is an orphan node, which is not attached to any AP. After nodes' deployment, the MN starts broadcasting beacons (Discovery Phase), implemented as *BeaconTimer* event in TinyOS. Two flags are responsible for defining the status of the node within Discovery Phase and Data Transmission Phase; namely *DiscoveryPhase* and *DataPhase*, which are set to TRUE and FALSE respectively at the start of the test. This setting enters the mobile node into the *BeaconTimer* (after calling the Boot event). In the Discovery Phase of the enhanced smart-HOP algorithm, *ws* beacons are sent in burst with the time interval of 10 ms, but the timer expires 100 ms after sending the last beacon. The intuition is to enable receiving at most 9 reply packets from the neighboring APs (modulo operator=10) within this period. However, in the Data Transmission Phase, only 10 ms is enough to receive the reply packet from the corresponding AP.

The *DataTimer* is responsible for the unicast data communication with the serving AP. The MN keeps assessing the current link within each *ws* by comparing the ARSSI

with T_l. If the current link quality is poor, the *DataTimer* stops (the starting moment of hand-off) and then the algorithm increases the number of hand-offs by one, changes the flags' status, and resumes the *BeaconTimer*.

At the AP side, the main tasks are performed at the *receive event* as we assume that they are activated when receiving a packet from a MN. Each time after receiving a packet, some checking should be done.

– Is the received packet sent from a MN device?

– Does the packet sent during a Discovery Phase or Data Transmission Phase?

– Is the AP in the middle of a communication or it is not serving any MN?

– Check the sequence number of the packets, which changes the replying slot.

Each group of packets in a sliding window has same sequence number with counters from 1 to the number of *ws*. The sequence number is advanced in the next sliding window. We also consider some timers which prevent possible failures during a communication. For instance, the AP may not receive all the packets within a window size. Instead of waiting for all packets, it starts various timers after receiving each packet. Immediately after receiving a packet at the AP, a timer is called, firing after a predefined amount of time. Before ending this waiting period, if the AP receives a packet, the timer stops and if it does not receive, a reply packet is sent back to the requested MN.

A.3 Packet format

Considering small memories, weak processors, limited energy, and limited packet size of low-power nodes, we devised a very careful and light algorithm with the requirement of additional 10 bytes information for the hand-off process. Figure A.1 depicts the packet format of smart-HOP, which is embedded to the beacon and data packets. There are five major fields which are defined as; *(i)* source (src) that reveals the *id* of a node, *(ii)* status (0 or 1) that defines the Discovery and Data Transmission Phases respectively, *(iii)* sequence number (seq_no) that is assigned to each burst of beacons and data packets within a *ws*,

(iv) counter that is assigned to each single packet in a burst within a *ws* and counts from 1 to *ws*, and *(v)* ARSSI that represents the average RSSI of the latest burst received at the corresponding AP.

There is always a possibility of losing some packets that we detect them by sniffers. For the sake of collecting all packets, we parsed packets from the MN through a serial port. The packet format is shown in Figure A.1, which has 21 bytes size. The number of hand-offs and the delay are calculated at the MN side. The *id*s of the sender and receiver are also stored for each packet.

AP/MN packet format	src 16 bits	status 16 bits	seq_no 32 bits	counter 8 bits	ARSSI 16 bits

Packet to serial port	MN_id 16 bits	AP_id 16 bits	status 8 bits	time 32 bits	seq_no 32 bits	counter 8 bits	ARSSI 16 bits	no_ho 8 bits	delay 32 bits

Figure A.1: Packet formats for communication and logging in TinyOS implementation.

Appendix B

Contiki implementation

This appendix outlines the implementation of the mRPL and mRPL+ mechanisms in Contiki operating system.

As we previously mentioned, we used three main control messages of RPL for triggering the hand-off mechanism, which are DIS, DIO and DAO control messages. We enhanced the packet format of DIS and DIO messages rather than creating new ones to obtain backward compatibility.

The enhanced DIO packet format is shown in Figure B.1, where the additional fields are colored and embolden. Please find the details on each field below.

- *RPLinstanceID*. It is an 8-bit field and shows the RPL Instance the DODAG belongs to.
- *Version Number*. It is an 8-bit field and shows the freshness of the DODAG.
- *Rank*. It is a 16-bit field showing the rank of the node sending the DIO.
- *Grounded*. It is a one big flag and is set when the root of the DODAG is connected to a non-LLN network or the public internet.
- *O*. It is set to 0.
- *Mode of Operation (MOP)*. It is a 3-bit field and indicates one of the four defined modes of operation of RPL: (i) 0: no downward routes maintained by RPL, (ii) 1:

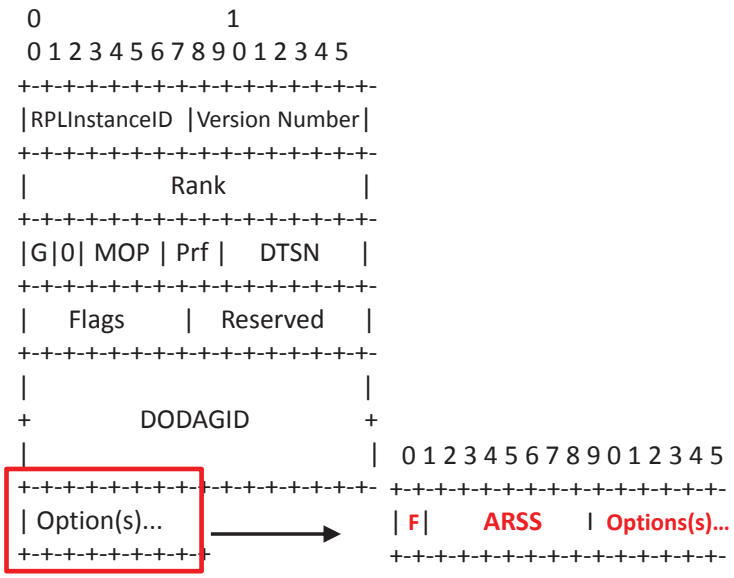


Figure B.1: Enhanced DIO control message packet format in mRPL and mRPL+.

non storing mode, (iii) 2: storing without multi-cast support and (iv) 3: storing with multi-cast support).

– *DODAG Preference*. The default value is 0. It is set by root to report its preference over other DODAG roots in the RPL instance.

– *Destination Advertisement Trigger Sequence Number (DTSN)*. It is an 8-bit field set by the node issuing the DIO message. The Destination Advertisement Trigger Sequence Number (DTSN) flag is used as part of the procedure to maintain downward routes.

– *Flags*. It is reserved for flags, initialized to 0 by sender and ignored by receiver.

– *Reserved*. It is 8-bit unused field, initialized to 0 by sender and ignored by receiver.

– *DODAGID*. It is 128-bit field, which uniquely identifies the DODAG and is set to the IPv6 address of the root.

– ***F***. This additional two bits distinguish three cases: (i) F=0 corresponds to the RPL DIO, (ii) F=1 indicates the mRPL DIO within the *Data Transmission Phase*, and (iii) F=2 reflects the mRPL DIO within the *Discovery Phase*.

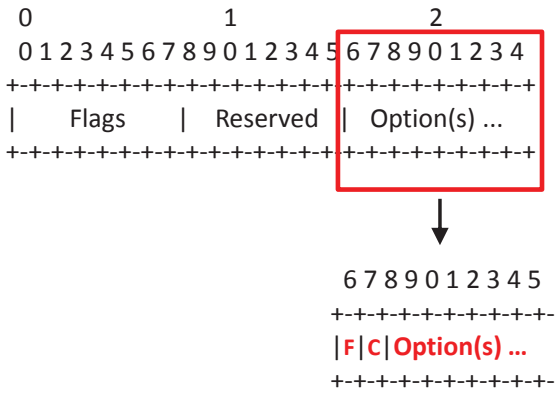


Figure B.2: Enhanced DIS control message packet format in mRPL and mRPL+.

– ***ARSSI***. This additional field holds the average RSSI reading at the potential parent node.

– *Options*. This field contains implementation specific values. In Contiki it may contains metric container type and flags, DIO minimum interval and doublings, minimum and maximum hop rank increment etc. It may also carry node prefix information.

The enhanced DIS packet format is shown in Figure B.2, where the additional fields are colored and embolden. Please find the details on the additional fields below.

– ***F***. This field distinguishes between the mRPL/mRPL+ DIS and the native RPL DIS. Initializing this field to "0" represents the multicast transmission of the RPL DIS. Instead, setting this field to "1" reflects the unicast mRPL DIS transmission.

– ***C***. This additional two bits describes the counter of DIS messages within a window size. In mRPL/mRPL+ with $ws = 3$, the counter increments to a maximum of 3.

The DAO packet format (which is unchanged in mRPL and mRPL+) is shown in Figure B.3. The DAO packet is transmitted after a successful hand-off process in order to establish upward path to enable data transmission from root to the mobile node. Please find the details on each field below.

– *K*. The 'K' flag indicates that the recipient of the DAO message is expected to send back an acknowledgement.

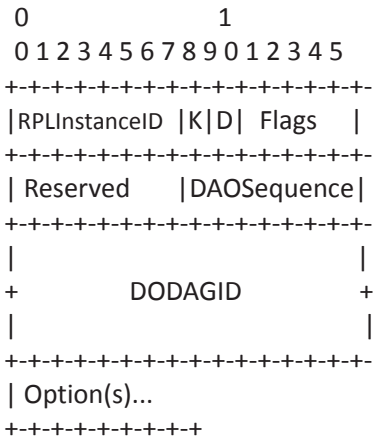


Figure B.3: The native DAO control message packet format in RPL, which is used in mRPL and mRPL+.

– *D*. This filed indicates that a local RPL instance ID is in use. DAOSequence: It is a counter incremented each time a new DAO message is sent by the node.

– *DAOSequence*. Incremented at each unique DAO message from a node and echoed in the DAO-ACK message.

– *DODAGID (Optional)*. It is 128-bit field, which uniquely identifies the DODAG. This field is only present when the 'D' flag is set.

Bibliography

[AAH11] Theodoros Anagnostopoulos, Christos Anagnostopoulos, and Stathes Hadjiefthymiades. An adaptive location prediction model based on fuzzy control. Elsevier Journal of Computer Communications, 34, 2011.

[ABCG11] Giuseppe Anastasi, Eleonora Borgia, Marco Conti, and Enrico Gregori. A hybrid adaptive protocol for reliable data delivery in wsns with multiple mobile sinks. The Computer Journal, 54(2):213–229, 2011.

[ABD10] N. Alam, A.T. Balaie, and A.G. Dempster. Dynamic path loss exponent and distance estimation in a vehicular network using doppler effect and received signal strength. In Vehicular Technology Conference (VTC). IEEE, 2010.

[ACDF09] Giuseppe Anastasi, Marco Conti, and Mario Di Francesco. Reliable and energy-efficient data collection in sparse sensor networks with mobile elements. Performance Evaluation, 66(12):791–810, 2009.

[AKBD06] Hüseyin Akcan, Vassil Kriakov, Hervé Brönnimann, and Alex Delis. Gps-free node localization in mobile wireless sensor networks. In Proceedings of the 5th ACM international workshop on Data engineering for wireless and mobile access, pages 35–42. ACM, 2006.

[AKJ10] Nadeem Ahmed, Salil Kanhere, and Sanjay Jha. Experimental evaluation of multi-hop routing protocols for wireless sensor networks. In Proceedings of the 9th ACM/IEEE International Conference on

Information Processing in Sensor Networks, Information Processing in Sensor Networks (IPSN), pages 416–417, New York, NY, USA, 2010. IEEE/ACM.

[All06] ZigBee Alliance. Zigbee specification. http://zigbee.org/, 2006.

[ASU05] Muneeb Ali, Tashfeen Suleman, and Zartash Afzal Uzmi. Mmac: A mobility-adaptive, collision-free mac protocol for wireless sensor networks. In Performance, Computing, and Communications Conference, 2005. IPCCC 2005. 24th IEEE International, pages 401–407. IEEE, 2005.

[BGLA03] Lichun Bao and Jose Juaquin Garcia-Luna-Aceves. Topology management in ad hoc networks. In Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing, pages 129–140. ACM, 2003.

[BGM+10] Adrian Burns, Barry R Greene, Michael J McGrath, Terrance J O'Shea, Benjamin Kuris, Steven M Ayer, Florin Stroiescu, and Victor Cionca. Shimmer^{TM}–a wireless sensor platform for noninvasive biomedical research. Sensors Journal, IEEE, 10(9):1527–1534, 2010.

[BJR+10] N. Baccour, M.B. Jamâa, D. Rosário, A. Koubâa, M. Alves, L.B. Becker, H. Youssef, and H. Fotouhi. Demo abstract: Radiale: a framework for benchmarking link quality estimators. In European Conference on Wireless Sensor Networks (EWSN), 2010.

[BKN+13] Nouha Baccour, Anis Koubâa, Claro Noda, Hossein Fotouhi, Mário Alves, Habib Youssef, Marco Zuniga, Carlo Boano, Kay Roemer, Daniele Puccinelli, Thiemo Voigt, and Luca Mottola. Radio link quality estimation in low-power wireless networks. Springer Lecture Notes in Electrical and Computer Engineering, in press, 2013.

[BKY+10] Nouha Baccour, Anis Koubâa, Habib Youssef, Maissa Ben Jamâa, Denis do Rosário, Mário Alves, and Leandro Becker. F-lqe: A fuzzy link

quality estimator for wireless sensor networks. In European Conference on Wireless Sensor Networks (EWSN). Springer, 2010.

[BKYA14] Nouha Baccour, Anis Koubaa, Habib Youssef, and Mario Alves. Reliable link quality estimation in low-power wireless networks and its impact on tree-routing. Ad Hoc Networks, 2014.

[BRKY09] Gargi Bag, Muhammad Taqi Raza, Ki-Hyung Kim, and Seung-Wha Yoo. Lowmob: intra-pan mobility support schemes for 6lowpan. Sensors, 9(7):5844–5877, 2009.

[BVD+09] C.A. Boano, T. Voigt, A. Dunkels, F. Osterlind, N. Tsiftes, L. Mottola, and P. Suarez. Exploiting the lqi variance for rapid channel quality assessment. In Proceedings of the 2009 International Conference on Information Processing in Sensor Networks, pages 369–370. IEEE, 2009.

[BVN+11] C.A. Boano, T. Voigt, C. Noda, K. Romer, and M. Zuniga. Jamlab: Augmenting sensornet testbeds with realistic and controlled interference generation. In Information Processing in Sensor Networks (IPSN). IEEE/ACM, 2011.

[CB04] Joe Capka and Raouf Boutaba. Mobility prediction in wireless networks using neural networks. In Management of Multimedia Networks and Services, pages 320–333. Springer, 2004.

[CBD02] Tracy Camp, Jeff Boleng, and Vanessa Davies. A survey of mobility models for ad hoc network research. Wireless communications and mobile computing, 2(5):483–502, 2002.

[CDL08] Xiaojuan Chao, Waltenegus Dargie, and Guan Lin. Energy model for h2s monitoring wireless sensor network. In Computational Science and Engineering, 2008. CSE'08. 11th IEEE International Conference on, pages 402–409. IEEE, 2008.

[CHP11] Thomas Clausen, Ulrich Herberg, and Matthias Philipp. A critical evaluation of the ipv6 routing protocol for low power and lossy networks (rpl). In 7th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), pages 365–372. IEEE, 2011.

[CIC+09] Aminul Haque Chowdhury, Muhammad Ikram, Hyon-Soo Cha, Hassen Redwan, S. M. Saif Shams, Ki-Hyung Kim, and Seung-Wha Yoo. Route-over vs mesh-under routing in 6lowpan. In Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly, IWCMC '09, pages 1208–1212. ACM, 2009.

[CJA+03] Thomas Clausen, Philippe Jacquet, Cédric Adjih, Anis Laouiti, Pascale Minet, Paul Muhlethaler, Amir Qayyum, Laurent Viennot, et al. Optimized link state routing protocol (olsr). RFC Editor, 2003.

[CJC09] Sunghyun Cho, E.W. Jang, and J.M. Cioffi. Handover in multihop cellular networks. IEEE Communications Magazine, 47(7), 2009.

[CKMT02] Tak-Ming Chan, Sam Kwong, Kim-Fung Man, and Kit-Sang Tang. Hard handoff minimization using genetic algorithms. Signal Processing, 82(8):1047–1058, 2002.

[CKPK+12] Mingang Cheng, Masako Kanai-Pak, Noriaki Kuwahara, Hiromi Itoh Ozaku, Kiyoshi Kogure, and Jun Ota. Dynamic scheduling–based inpatient nursing support: applicability evaluation by laboratory experiments. International Journal of Autonomous and Adaptive Communications Systems, 5(1):39–57, 2012.

[CLBR10a] Octav Chipara, Chenyang Lu, Thomas C. Bailey, and Gruia-Catalin Roman. Reliable clinical monitoring using wireless sensor networks: Experiences in a step-down hospital unit. In Embedded Networked Sensor Systems (SenSys). ACM, 2010.

[CLBR10b] Octav Chipara, Chenyang Lu, Thomas C. Bailey, and Gruia-Catalin Roman. Reliable clinical monitoring using wireless sensor networks: experiences in a step-down hospital unit. In Embedded Networked Sensor Systems (SenSys). ACM, 2010.

[CMN14] Cosmin Cobarzan, Julien Montavont, and Thomas Noel. Analysis and performance evaluation of rpl under mobility. In Symposium on Computers and Communication (ISCC), pages 1–6. IEEE, 2014.

[Com14a] European Commission. Factories of the future 2020: multi-annual roadmap for the contractual ppp under horizon 2020, research roadmap produced by the european factories of the future research association (effra). http://www.effra.eu/attachments/article/129/FactoriesDecember 2014.

[Com14b] European Commission. flexware : Flexible wireless automation in real-time environments. http://www.flexware.at/, December 2014.

[CPR+08] Tiago Camilo, Pedro Pinto, André Rodrigues, Jorge Sá Silva, and Fernando Boavida. Mobility management in ip-based wireless sensor networks. In World of Wireless, Mobile and Multimedia Networks, 2008. WoWMoM 2008. 2008 International Symposium on a, pages 1–8. IEEE, 2008.

[CT14] Inc. Crossbow Technology. Telosb datasheet. http://www.willow.co.uk, December 2014.

[DCABM03] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A high-throughput path metric for multi-hop wireless routing. In Proceedings of the 9th Annual International Conference on Mobile Computing and Networking, pages 134–146. ACM, 2003.

[DD13] Qian Dong and Waltenegus Dargie. A survey on mobility and mobility-aware mac protocols in wireless sensor networks. IEEE Communications Surveys and Tutorials, 15(1):88–100, 2013.

[DDB12] Sébastien Dawans, Simon Duquennoy, and Olivier Bonaventure. On link estimation in dense rpl deployments. In 7th IEEE International Workshop on Practical Issues in Building Sensor Network Applications, 2012.

[DFDA11] Mario Di Francesco, Sajal K Das, and Giuseppe Anastasi. Data collection in wireless sensor networks with mobile elements: A survey. ACM Transactions on Sensor Networks (TOSN), 8(1):7, 2011.

[DGV04] Adam Dunkels, Bjorn Gronvall, and Thiemo Voigt. Contiki-a lightweight and flexible operating system for tiny networked sensors. In Local Computer Networks (LCN). IEEE, 2004.

[DNW07] Serhan Dagtas, Yuri Natchetoi, and Huaigu Wu. An integrated wireless sensing and mobile processing architecture for assisted living and healthcare applications. In 1st ACM SIGMOBILE international workshop on Systems and networking support for healthcare and assisted living environments, pages 70–72. ACM, 2007.

[DRS+05] Karthik Dantu, Mohammad Rahimi, Hardik Shah, Sandeep Babel, Amit Dhariwal, and Gaurav S Sukhatme. Robomote: enabling mobility in sensor networks. In Proceedings of the 4th international symposium on Information processing in sensor networks, page 55. IEEE Press, 2005.

[DS09] Karthik Dantu and G Sukhatme. Connectivity vs. control: Using directional and positional cues to stabilize routing in robot networks. In Robot Communication and Coordination, 2009. ROBOCOMM'09. Second International Conference on, pages 1–6. IEEE, 2009.

[Dun11] Adam Dunkels. The contikimac radio duty cycling protocol. SICS Technical Report, 2011, 2011.

[FAKB10] H. Fotouhi, M. Alves, A. Koubaa, and N. Baccour. On a reliable hand-off procedure for supporting mobility in wireless sensor networks. In Workshop on Real-Time Networks (RTN), 2010.

[FAKZ11] H. Fotouhi, M. Alves, A. Koubâa, and M. Zuniga. Poster abstract: Smart-hop: A reliable handoff procedure for supporting mobility in wireless sensor networks. In European Conference on Wireless Sensor Networks (EWSN). Springer, 2011.

[FAS14] C. Fischione, G. Athanasiou, and F. Santucci. Dynamic optimization of generalized least squares handover algorithms. Wireless Communications, IEEE Transactions on, 13(3):1235–1249, March 2014.

[FAZK14] H. Fotouhi, M. Alves, M.Z. Zamalloa, and A Koubaa. Reliable and fast hand-offs in low-power wireless networks. IEEE Transactions on Mobile Computing, 13(11):2620–2633, Nov 2014.

[FFL06] David Johnson Tim Stack Russ Fish, Daniel Montrallo Flickinger, and Leigh Stoller Robert Ricci Jay Lepreau. Mobile emulab: A robotic wireless and sensor network testbed. In IEEE Conference on Computer Communications (INFOCOM). IEEE, 2006.

[FGJ+06] Rodrigo Fonseca, Omprakash Gnawali, Kyle Jamieson, Sukun Kim, Philip Levis, and Alec Woo. The collection tree protocol (ctp). TinyOS TEP, 123:2, 2006.

[FGJL07] R. Fonseca, O. Gnawali, K. Jamieson, and P. Levis. Four-bit wireless link estimation. In Proceedings of the Sixth Workshop on Hot Topics in Networks (HotNets VI), volume 2007, 2007.

[Fis07] Reed Fisher. 60 ghz wpan standardization within ieee 802.15. 3c. In Signals, Systems and Electronics, 2007. ISSSE'07. International Symposium on, pages 103–105. IEEE, 2007.

[FMA14a] Hossein Fotouhi, Daniel Moreira, and Mário Alves. Mobile iot: smart-hop over rpl. CISTER Research Center Technical Report, 2014.

[FMA14b] Hossein Fotouhi, Daniel Moreira, and Mário Alves. mrpl: boosting mobility in the internet of things. Elsevier Journal on Ad-hoc Networks, 26:17–35, 2014.

[FMA15] Hossein Fotouhi, Daniel Moreira, and Mário Alves. mrpl+: a mobility management framework in rpl. Submitted to an International Journal, 2015.

[FZA+12] Hossein Fotouhi, Marco Zuniga, Mario Alves, Anis Koubaa, and Pedro Marrón. Smart-hop: A reliable handoff mechanism for mobile wireless sensor networks. In European Conference on Wireless Sensor Networks (EWSN). Springer, 2012.

[GBE06] Yaoyao Gu, Doruk Bozdag, and Eylem Ekici. Mobile element based differentiated message delivery in wireless sensor networks. In Proceedings of the 2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks, pages 83–92. IEEE Computer Society, 2006.

[GFJ+09] Omprakash Gnawali, Rodrigo Fonseca, Kyle Jamieson, David Moss, and Philip Levis. Collection tree protocol. In Conference on Embedded Networked Sensor Systems (SenSys), pages 1–14. ACM, 2009.

[GIN14] Ginseng: performance control in wireless sensor networks. http://www.ucc.ie/en/misl/research/previous/ginseng/, December 2014.

[GT00] H Gan and B Treister. Adaptive frequency hopping implementation proposals for ieee 802.15.1/2 wpan. IEEE working group, 2000.

[GZD11] Shuai Gao, Hongke Zhang, and Sajal K Das. Efficient data collection in wireless sensor networks with path-constrained mobile sinks. Mobile Computing, IEEE Transactions on, 10(4):592–608, 2011.

[Haa00] Jaap C Haartsen. The bluetooth radio system. Personal Communications, IEEE, 7(1):28–36, 2000.

[HC08] J.W. Hui and D.E. Culler. Extending ip to low-power, wireless personal area networks. Internet Computing, IEEE, 12(4):37–45, July 2008.

[HCB02] Wendi B Heinzelman, Anantha P Chandrakasan, and Hari Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. IEEE Transactions on Wireless Communications, 1(4):660–670, 2002.

[HE04] Lingxuan Hu and David Evans. Localization for mobile sensor networks. In Proceedings of the 10th annual international conference on Mobile computing and networking, pages 45–57. ACM, 2004.

[HFB09] Jerome Harri, Fethi Filali, and Christian Bonnet. Mobility models for vehicular ad hoc networks: a survey and taxonomy. Communications Surveys & Tutorials, IEEE, 11(4):19–41, 2009.

[HKB99] Wendi Rabiner Heinzelman, Joanna Kulik, and Hari Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In 5th annual ACM/IEEE international conference on Mobile computing and networking, pages 174–185. IEEE/ACM, 1999.

[HS06] Zygmunt J Haas and Tara Small. A new networking model for biological applications of ad hoc sensor networks. IEEE/ACM Transactions on Networking, 14(1):27–40, 2006.

[Hun06] Nick Hunn. An introduction to wibree. http://www.ezurio.com, 2006.

[(IB14] Cisco's Internet Business Solutions Group (IBSG). The internet of things. http://share.cisco.com/internet-of-things.html, December 2014.

[IGE+03] Chalermek Intanagonwiwat, Ramesh Govindan, Deborah Estrin, John Heidemann, and Fabio Silva. Directed diffusion for wireless sensor networking. Networking, IEEE/ACM Transactions on, 11(1):2–16, 2003.

[Inc14] MEMSIC Inc. Iris data sheet. http://www.memsic.com, December 2014.

[Ins07] Texas Instruments. Cc2420 datasheet. http://www.ti.com/lit/ds/symlink/cc2420.pdf, 2007.

[ISA09] ISA100 ISA. 100.11 a-2009: Wireless systems for industrial automation: Process control and related applications. International Society of Automation: Research Triangle Park, NC, USA, 2009.

[JAZ05] Hyewon Jun, Mostafa H Ammar, and Ellen W Zegura. Power management in delay tolerant networks: a framework and knowledge-based mechanisms. In Sensor and Ad Hoc Communications and Networks (SECON), volume 5, pages 418–429. IEEE, 2005.

[JEH+08] G. Jakllari, S. Eidenbenz, N. Hengartner, S.V. Krishnamurthy, and Michalis Faloutsos. Link positions matter: A noncommutative routing metric for wireless mesh network. In IEEE Conference on Computer Communications (INFOCOM), pages 744–752, April 2008.

[JKK+11] Raja Jurdak, Kevin Klues, Brano Kusy, Christian Richter, Koen Langendoen, and Michael Brunig. Opal: A multiradio platform for high throughput wireless sensor networks. Embedded Systems Letters, IEEE, 3(4):121–124, 2011.

[JMB+01] David B Johnson, David A Maltz, Josh Broch, et al. Dsr: The dynamic source routing protocol for multi-hop wireless ad hoc networks. Elsevier Journal on Ad-hoc Networks, 5, 2001.

[JOW+02] Philo Juang, Hidekazu Oki, Yong Wang, Margaret Martonosi, Li Shiuan Peh, and Daniel Rubenstein. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. In Sigplan Notices, volume 37, pages 96–107. ACM, ACM, 2002.

[JSB+06] Sushant Jain, Rahul C Shah, Waylon Brunette, Gaetano Borriello, and Sumit Roy. Exploiting mobility for energy efficient data collection in wireless sensor networks. Mobile Networks and Applications, 11(3):327–339, 2006.

[JSS+10] Antonio J Jara, Ricardo M Silva, Jorge Silva, MA Zamora-Izquierdo, and AFG Skarmeta. Mobile ipv6 over wireless sensor networks (6lowpan) issues and feasibility. European Wireless Sensor Networks, 2010.

[KBBAS12] I.E. Korbi, M. Ben Brahim, C. Adjih, and L.A. Saidane. Mobility enhanced rpl for wireless sensor networks. In Network of the Future (NOF), 2012 Third International Conference on the, pages 1–8, Nov 2012.

[KC14] J. Ko and M. Chang. Momoro: Providing mobility support for low-power wireless applications. Systems Journal, IEEE, PP(99):1–10, 2014.

[KL07] George K Karagiannidis and Athanasios S Lioumpas. An improved approximation for the gaussian q-function. IEEE Communications Letters, 11(8):644–646, 2007.

[KSJ+04a] A. Kansal, A.A. Somasundara, D.D. Jea, M.B. Srivastava, and D. Estrin. Intelligent fluid infrastructure for embedded networks. In Proceedings of the 2nd international conference on Mobile systems, applications, and services, pages 111–124. ACM, 2004.

[KSJ+04b] Aman Kansal, Arun A Somasundara, David D Jea, Mani B Srivastava, and Deborah Estrin. Intelligent fluid infrastructure for embedded networks. In Proceedings of the 2nd international conference on Mobile systems, applications, and services, pages 111–124. ACM, 2004.

[KWS+10] JeongGil Ko, Qiang Wang, Thomas Schmid, Wanja Hofer, Prabal Dutta, and Andreas Terzis. Egs: A cortex m3-based mote platform. In Sensor Mesh and Ad Hoc Communications and Networks (SECON), pages 1–3. IEEE, 2010.

[LFNS12] Xu Li, R. Falcon, A. Nayak, and I. Stojmenovic. Servicing wireless sensor networks by mobile robots. IEEE Communication Magazine, 50(7):147–154, 2012.

[LH05] Jun Luo and J-P Hubaux. Joint mobility and routing for lifetime elongation in wireless sensor networks. In IEEE Conference on Computer Communications (INFOCOM), volume 3, pages 1735–1746. IEEE, 2005.

[LMFJ+04a] K. Lorincz, D.J. Malan, T.R.F. Fulford-Jones, A. Nawoj, A. Clavel, V. Shnayder, G. Mainland, M. Welsh, and S. Moulton. Sensor networks for emergency response: challenges and opportunities. IEEE Trans. Pervasive Computing, 3(4), 2004.

[LMFJ+04b] Konrad Lorincz, David J Malan, Thaddeus RF Fulford-Jones, Alan Nawoj, Antony Clavel, Victor Shnayder, Geoffrey Mainland, Matt Welsh, and Steve Moulton. Sensor networks for emergency response: challenges and opportunities. Pervasive Computing, IEEE, 3(4):16–23, 2004.

[LMP+04] Philip Levis, Sam Madden, Joseph Polastre, Robert Szewczyk, Alec Woo, David Gay, Jason Hill, Matt Welsh, Eric Brewer, and David Culler. Tinyos: An operating system for sensor networks. In in Ambient Intelligence. Springer Verlag, 2004.

[LPCS04] Philip Alexander Levis, Neil Patel, David Culler, and Scott Shenker. Trickle: A self regulating algorithm for code propagation and maintenance in wireless sensor networks, 2004. In Proceedings of the 1st Conference on Symposium on Networked Systems Design and Implementation - Volume 1. USENIX Association, 2004.

[LPP+06] Jun Luo, Jacques Panchard, Michał Piórkowski, Matthias Grossglauser, and Jean-Pierre Hubaux. Mobiroute: Routing towards a mobile sink for improving lifetime in sensor networks. In Distributed Computing in Sensor Systems, pages 480–497. Springer, 2006.

[LR02] Stephanie Lindsey and Cauligi S Raghavendra. Pegasis: Power-efficient gathering in sensor information systems. In Aerospace conference proceedings, 2002. IEEE, volume 3, pages 3–1125. IEEE, 2002.

[LSD+12] K.C. Lee, R. Sudhaakar, L. Dai, S. Addepalli, and M. Gerla. Rpl under mobility. In Consumer Communications and Networking Conference (CCNC), 2012 IEEE, pages 300–304, Jan 2012.

[LSN+12] Kevin C Lee, Raghuram Sudhaakar, Jianxia Ning, Lillian Dai, Sateesh Addepalli, JP Vasseur, and Mario Gerla. A comprehensive evaluation of rpl under mobility. International Journal of Vehicular Technology, 2012, 2012.

[LYC+05] Haiyun Luo, Fan Ye, Jerry Cheng, Songwu Lu, and Lixia Zhang. Ttdd: Two-tier data dissemination in large-scale wireless sensor networks. Wireless Networks, 11(1-2):161–175, 2005.

[MdDJGSBO14] JoseRamiro Martinez-de Dios, Adrian Jimenez-Gonzalez, Alberto San Bernabe, and Anibal Ollero. Conet integrated testbed architecture. In A Remote Integrated Testbed for Cooperating Objects, SpringerBriefs in Electrical and Computer Engineering, pages 23–39. This work

has been carried out within the Cooperating Objects European Network of Excellence http://www.cooperating–objects.eu/. Springer International Publishing, 2014.

[MDT08] B.B. Madan, S. Dharmaraja, and K.S. Trivedi. Combined guard channel and mobile-assisted handoff for cellular networks. IEEE Transactions on Vehicular Technology, 57(1), 2008.

[MHT00] Yue Ma, J.J. Han, and K.S. Trivedi. Call admission control for reducing dropped calls in code division multiple access (cdma) cellular systems. In IEEE Conference on Computer Communications (INFOCOM). IEEE, 2000.

[MKHC07] Gabriel Montenegro, Nandakishore Kushalnagar, J Hui, and D Culler. Transmission of ipv6 packets over ieee 802.15. 4 networks. Internet proposed standard RFC, 4944, 2007.

[MMK12] Pedro Jos Marron, Daniel Minder, and Stamatis Karnouskos. The Emerging Domain of Cooperating Objects. Springer, 2012.

[MRN14] Julien Montavont, Damien Roth, and Thomas Noël. Mobile ipv6 in internet of things: Analysis, experimentations and optimizations. Ad Hoc Networks, 14:15–25, March 2014.

[MSA03] Arunesh Mishra, Minho Shin, and William Arbaugh. An empirical analysis of the ieee 802.11 mac layer handoff process. SIGCOMM, 2003.

[Mul07] Geoff Mulligan. The 6lowpan architecture. In Proceedings of the 4th workshop on Embedded networked sensors. ACM, 2007.

[NNS07] T Narten, E Nordmark, and W Simpson. H. soliman," neighbor discovery for ip version 6 (ipv6). Technical report, RFC 4861, September, 2007.

[NSNS98] Thomas Narten, William Allen Simpson, Erik Nordmark, and Hesham Soliman. Neighbor discovery for ip version 6 (ipv6). RFC 2461, December 1998.

[Ost14] Fredrik Osterlind. Mobility cooja plugin. https://github.com/contiki-os/contiki/wiki, December 2014.

[oT14] Massachusetts Institute of Technology. irobot. http://www.irobot.com/About-iRobot.aspx, December 2014.

[par14] Workshop participants. Final report from the nsf workshop on future directions in wireless networking. http://ecedha.org/docs/nsf-nets/final-report.pdf?sfvrsn=0, November 2014.

[PBRD03] Charles Perkins, E Belding-Royer, and Samir Das. Ad hoc on demand distance vector (aodv) routing (rfc 3561). IETF MANET Working Group, 2003.

[PJKC05] Sangheon Pack, Hakyung Jung, Taekyoung Kwon, and Yanghee Choi. Snc: a selective neighbor caching scheme for fast handoff in ieee 802.11 wireless networks. SIGMOBILE Mob. Comput. Commun. Rev., 9, 2005.

[PK11] J. Petajajarvi and H. Karvonen. Soft handover method for mobile wireless sensor networks based on 6lowpan. In Distributed Computing in Sensor Systems (DCOSS). IEEE, 2011.

[PM11] Yuanteng Pei and Matt W. Mutka. Stars: Static relays for multi-robot real-time search and monitoring. In Distributed Computing in Sensor Systems (DCOSS). IEEE, 2011.

[pm14] FASyS project members. Fasys: Absolutely safe and healthy factory. http://www.fasys.es/en/proyecto.php, December 2014.

[PR99] Charles E Perkins and Elizabeth M Royer. Ad-hoc on-demand distance vector routing. In IEEE Workshop on Mobile Computing Systems and Applications. IEEE, 1999.

[Pro14a] MASQOTS Project. mrpl implementation in contiki. http://www.cister.isep.ipp.pt/masqots/sources.html, December 2014.

[Pro14b] MASQOTS Project. mrpl+ implementation in contiki. http://www.cister.isep.ipp.pt/masqots/sources.html, December 2014.

[Pro14c] MASQOTS Project. smart-hop implementation in tinyos. http://www.cister.isep.ipp.pt/masqots/sources.html, December 2014.

[PSH+02] Michal Pióro, Áron Szentesi, János Harmatos, Alpár Jüttner, Piotr Gajowniczek, and Stanislaw Kozdrowski. On open shortest path first related network optimisation problems. Performance Evaluation, 48(1), 2002.

[Rap96] T.S. Rappaport. Wireless communications: principles and practice. IEEE, 1996.

[RHG+11] O. Rensfelt, F. Hermans, P. Gunningberg, L.Å. Larzon, and E. Björnemo. Repeatable experiments with mobile nodes in a relocatable wsn testbed. The Computer Journal, 54(12), 2011.

[RMN12] Damien Roth, Julien Montavont, and Thomas Noel. Performance evaluation of mobile ipv6 over 6lowpan. In Proceedings of the 9th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks, PE-WASUN '12, pages 77–84. ACM, 2012.

[Rob06] Chris M Roberts. Radio frequency identification (rfid). Computers & Security, 25(1):18–26, 2006.

[RS05] I. Ramani and S. Savage. Syncscan: practical fast handoff for 802.11 infrastructure networks. In IEEE Conference on Computer Communications (INFOCOM), 2005.

[RS09] Anjali Raja and Xiao Su. Mobility handling in mac for wireless ad hoc networks. Wireless Communications and Mobile Computing, 9(3):303–311, 2009.

[RWB08] Jayanthi Rao, Tao Wu, and Subir Biswas. Network-assisted sink navigation protocols for data harvesting in sensor networks. In Wireless Communications and Networking Conference, 2008. WCNC 2008. IEEE, pages 2887–2892. IEEE, 2008.

[SAZ10] Lifeng Sang, Anish Arora, and Hongwei Zhang. On link asymmetry and one-way estimation in wireless sensor networks. ACM Transactions on Sensor Networks (TOSN), 6(2):12, 2010.

[SB11] Zach Shelby and Carsten Bormann. 6LoWPAN: The wireless embedded Internet, volume 43. John Wiley & Sons, 2011.

[Sch06] Christian Schindelhauer. Mobility in wireless networks. In Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM), pages 100–116. Springer, 2006.

[SCN11] Z Shelby, S Chakrabarti, and E Nordmark. Neighbor discovery optimization for low power and lossy networks (6lowpan). Work in progress, IETF draft-ietf-6lowpan-nd-18, 2011.

[SDTL10] Kannan Srinivasan, Prabal Dutta, Arsalan Tavakoli, and Philip Levis. An empirical study of low-power wireless. ACM Transactions on Sensor Networks (TOSN), 6(2):16, 2010.

[SF06] Tara Salih and Kemal Fidanboylu. Modeling and analysis of queuing handoff calls in single and two-tier cellular networks. Computer Communications, 29(17), 2006.

[SHM+08] Jianping Song, Song Han, Aloysius K Mok, Deji Chen, Mike Lucas, and Mark Nixon. Wirelesshart: Applying wireless technology in real-time industrial process control. In Real-Time and Embedded

Technology and Applications Symposium, 2008. RTAS'08. IEEE, pages 377–386. IEEE, 2008.

[SKAL08] Kannan Srinivasan, Maria A. Kazandjieva, Saatvik Agarwal, and Philip Levis. The b-factor: measuring wireless link burstiness. In Embedded Networked Sensor Systems (SenSys). ACM, 2008.

[SL06] Kannan Srinivasan and Philip Levis. Rssi is under appreciated. In Third Workshop on Embedded Networked Sensors (EmNets), 2006.

[SMA04] Minho Shin, Arunesh Mishra, and William A. Arbaugh. Improving the latency of 802.11 hand-offs using neighbor graphs. In Proceedings of the 2Nd International Conference on Mobile Systems, Applications, and Services, MobiSys. ACM, 2004.

[SRJB03] Rahul C Shah, Sumit Roy, Sushant Jain, and Waylon Brunette. Data mules: Modeling and analysis of a three-tier architecture for sparse sensor networks. Ad Hoc Networks, 1(2):215–233, 2003.

[SSB12] Ricardo Silva, Jorge Sa Silva, and Fernando Boavida. A proposal for proxy-based mobility in wsns. Elsevier Journal of Computer Communications, 35(10), 2012.

[SSB14] Ricardo Silva, Jorge Sá Silva, and Fernando Boavida. Mobility in wireless sensor networks - survey and proposal. Computer Communications, 52:1–20, 2014.

[STGS02] Curt Schurgers, Vlasios Tsiatsis, Saurabh Ganeriwal, and Mani Srivastava. Optimizing sensor networks in the energy-latency-density design space. Mobile Computing, IEEE Transactions on, 1(1):70–80, 2002.

[STS02] Curt Schurgers, Vlasios Tsiatsis, and Mani B Srivastava. Stem: Topology management for energy efficient sensor networks. In Aerospace Conference Proceedings, 2002. IEEE, volume 3, pages 3–1099. IEEE, 2002.

[TAFK12] Meriam Thamri, Mário Alves, Hossein Fotouhi, and Houda Khedher. Evaluation of smart-hop: a handoff approach for mobile wireless sensor networks. Master's thesis, Higher School of Communication of Tunis, Sup'Com, 2012.

[tea14] MetaGeek team. Wi-spy spectrum analyzer. http://www.metageek.net/products/wi-spy, December 2014.

[Wei10] Eric W Weisstein. CRC concise encyclopedia of mathematics. CRC press, 2010.

[Win12] Tim Winter. Rpl: Ipv6 routing protocol for low-power and lossy networks. IETF, 2012.

[WL97] D. Wong and Teng Joon Lim. Soft handoffs in cdma mobile systems. IEEE Personal Communications, 4(6), 1997.

[WMRD11] Thomas Watteyne, Antonella Molinaro, Maria Grazia Richichi, and Mischa Dohler. From manet to ietf roll standardization: A paradigm shift in wsn routing protocols. IEEE Communications Surveys & Tutorials, 13(4), 2011.

[WTC03] Alec Woo, Terence Tong, and David Culler. Taming the underlying challenges of reliable multihop routing in sensor networks. In Embedded Networked Sensor Systems (SenSys), 2003.

[WY05] Jie Wu and Shuhui Yang. Smart: A scan-based movement-assisted sensor deployment method in wireless sensor networks. In IEEE Conference on Computer Communications (INFOCOM), volume 4, pages 2313–2324. IEEE, 2005.

[YCLZ01] Fan Ye, Alvin Chen, Songwu Lu, and Lixia Zhang. A scalable solution to minimum cost forwarding in large sensor networks. In 10th International Conference on Computer Communications and Networks, pages 304–309. IEEE, 2001.

[YF04] Ossama Younis and Sonia Fahmy. Heed: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. Mobile Computing, IEEE Transactions on, 3(4):366–379, 2004.

[YKH14] Jeongbae Yun, Murad Khan, and Kijun Han. A fast handoff scheme for streaming service in wireless sensor networks. International Journal of Distributed Sensor Networks, 501:183802, 2014.

[YPMM01] M. Ylianttila, M. Pande, J. Makela, and P. Mahonen. Optimization scheme for mobile users performing vertical handoffs between ieee 802.11 and gprs/edge networks. In IEEE GLOBECOM. IEEE, 2001.

[YY06] Guang-Zhong Yang and Magdi Yacoub. Body sensor networks. Springer, 2006.

[ZAZ04] Wenrui Zhao, Mostafa Ammar, and Ellen Zegura. A message ferrying approach for data delivery in sparse mobile ad hoc networks. In 5th ACM international symposium on Mobile ad hoc networking and computing, pages 187–198. ACM, 2004.

[ZD10] Tang Zhiyong and Waltenegus Dargie. A mobility-aware medium access control protocol for wireless sensor networks. In GLOBECOM Workshops, pages 109–114. IEEE, 2010.

[ZIH+11] M. Zuniga, I. Irzynska, J. Hauer, T. Voigt, C.A. Boano, and K. Roemer. Link quality ranking: Getting the best out of unreliable links. In Distributed Computing in Sensor Systems (DCOSS). IEEE, 2011.

[ZK04] M. Zuniga and B. Krishnamachari. Analyzing the transitional region in low power wireless links. In Sensor and Ad Hoc Communications and Networks (SECON). IEEE, 2004.

[ZL04] Jianliang Zheng and Myung J Lee. A comprehensive performance study of ieee 802.15. 4, 2004.

[ZV11] Z. Zinonos and V. Vassiliou. S-ginmob: Soft-handoff solution for mobile users in industrial environments. In Distributed Computing in Sensor Systems (DCOSS). IEEE, 2011.

[ZVC14] Zinon Zinonos, Vasos Vassiliou, and Chrysostomos Chrysostomou. Adaptive fuzzy logic mobility management for wsn. In International Conference on Distributed Computing in Sensor Systems (DCOSS), pages 302–307. IEEE, 2014.